COMPUTER AIDED INTERNAL ARRANGEMENTS OF SHIPS

by

David Blair Murton

B.S. Stanford University

1955

Submitted in partial fulfillment

of the requirements

for the degree of Naval Engineer

and for the degree of Master of Science

in Naval Architecture and Marine Engineering

at the

Massachusetts Institute of Technology

June, 1965

COMPUTER AIDED INTERNAL ARRANGEMENTS OF SHIPS

by

David Blair Murton

B.S. Stanford University

1955

Submitted in partial fulfillment

of the requirements

for the degree of Naval Engineer

and for the degree of Master of Science

in Naval Architecture and Marine Engineering

at the

Massachusetts Institute of Technology

June, 1965

Signature of Author. . . . .

Department of Naval Architecture
and Marine Engineering, 21 May 1965

Certified by . . . .

Thesis Supervisor

Accepted by. . . . .

Chairman, Departmental Committee
on Graduate Students

COMPUTER AIDED INTERNAL ARRANGEMENTS OF SHIPS

by

David Blair Murton

Submitted to the Department of Naval Architecture and Marine Engineering on 21 May 1965 in partial fulfillment of the requirements for the degree of Naval Engineer and for the degree of Master of Science in Naval Architecture and Marine Engineering.

This is a progress report on the status of a system for preparing ship internal arrangements on the cathode ray tube of the ESL Display at Project MAC, Massachusetts Institute of Technology, Cambridge. It documents the computer programs used for the arrangements system and presents instructions for the use of the system.

Thesis Supervisor: Philip Mandel

Title: Professor of Naval Architecture and Marine Engineering

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The purpose of this thesis was to start work on a system for using a graphical input output device to aid the ship designer in arranging the equipment and functions of a ship. The motivation was twofold: 1) to make it possible to develop arrangement plans directly in three dimensions and 2) to make the process of altering an arrangement much more flexible and faster. Because of the trial and error nature of the arrangement process, it was felt that a considerable increase in the speed of preparing arrangements could be realized by providing the designer with a means for quickly altering a plan and evaluating the effect of the alteration. The specific goal was to develop a system for the ESL Display at Project MAC to enable Naval Architects at M.I.T. to develop ship arrangements on the display. It was proposed to write a set of computer routines to enable the designer to perform the following operations on the display:

1) Draw three-dimensional pictures made of lines lying only in the three orthogonal planes.[1]
2) Apply horizontal and vertical constraints to these lines.[1]
3) File pictures in a disc file in order to create a catalog of objects used in ship arrangements. These pictures were to be called by a name in order to be displayed on the scope.[2]
4) Assign and change the scale of pictures being displayed.[1]
5) Move lines already drawn.[3]
6) Rotate the picture displayed about any arbitrary point.[1]
7) Name pictures and picture parts and display written information on the scope.[2]
8) Delete pictures,and blank and restore lines.[2]
9) Connect pictures together to form a more complex picture.[1]
10) Break apart complex pitcures into their components for modification.[2]

---

[1] Computer routine is written in satisfactory form.

[2] Part of the routine is written or the facility can be demonstrated to some extent. More work needs to be done.

[3] No work has been accomplished.

11) Compute center of gravity and total weight based on the position of pictures and the weight data provided.[1]

12) Ask for volume and area information based on the pictures drawn.[3]

13) Get the distance between any two parts of a picture.[2]

This paper attempts to describe what has thus far been done on the computer-aided arrangement system so that anyone will be able to pick up the work at a later date and extend the programs and ideas. The paper first describes the ESL Display and the supporting software. Following this is a section telling how to use the system as developed so far. A description of the programs is then made which includes the uses of some of the variables and descriptive flow diagrams. The last section is a discussion of work yet to be done and some ideas on alternate methods of doing what has already been done.

II. HARDWARE AND SOFTWARE

ESL Console

The ESL Console is a graphical input output device built by the Electronic Systems Laboratory at M.I.T. It is located in Room 908 Technology Square, Cambridge. Massachusetts. It works through a data channel to the project MAC IBM 7094 computer as one of the time sharing consoles. Figure (1) is a schematic representation of the time sharing system as it applies to the ESL Console. Facilities available to the user are:

1 - A 10-inch cathode ray tube upon which pictures are displayed.

2 - A lightpen.

3 - One set of 9 decimal switches.

4 - Two banks of 36 toggle switches.

5 - Three coding knobs.

6 - One coding globe or joystick.

7 - A set of 36 push buttons.

Control of the display seen on the CRT is determined by a set of commands called a display list. The console performs various plotting functions in accordance with the commands it reads from the display list. Space in A Core of the 7094 computer is allocated for the storage of the display list. This space is 1100 contiguous 36 bit registers. The display list is read by the ESL Console once every 30 milliseconds to produce a picture on the CRT. Altering the commands in the display list alters the picture displayed. Programs which reside in A Core can alter the display list at a rate fast enough to make the display move in the manner of a motion picture image. B Core programs can only affect the display list at discrete times: whenever the time sharing system is serving the console. Computation for the lightpen tracking cross and multiplication of the display list vectors by a rotation matrix are done by the console hardware. None of the coding knobs or switches have any effect on the display except as their outputs are interpreted by computer programs in A Core or B Core. A list of some of the most frequently used commands and their format are shown in Figure (2). For more detail on all aspects of the ESL Console, see Reference 1.

[illegible]

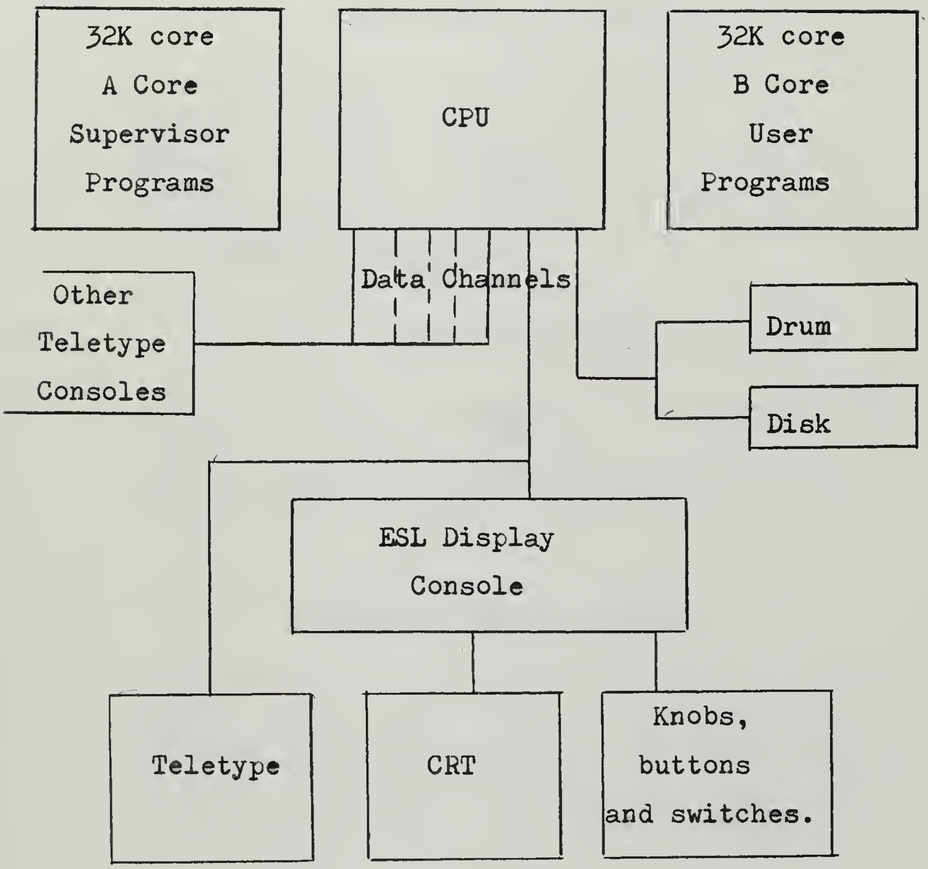


Figure (1)

Schematic of MAC Computer System

Line Generate Command

001	$\triangle Z^{*}$	000 or 100	
000	$\triangle X$		$\triangle Y$

*$\triangle Z$ word is optional

Setpoint Command

011	H	000 or 100	V

Rotation Matrix Command

101	i_h	j_h	k_h
110	i_v	j_v	k_v

Figure (2)

Often used console commands.

When considering pictures on the scope, it is convenient to think of two fields in which lines can be drawn: the scope field and the console field. The console field is 80 inches on a side while the scope field is only 10 inches. The scope field can be thought of as located anywhere in the console field the user wishes, dependent on the manner in which he manipulates the displayed picture. The coordinates of the console and scope fields are called h for horizontal, v for vertical and d for depth. Pictures as they exist in the display list have their own right handed coordinate system whose coordinate names are x, y and z. With no rotation by a rotation matrix in the console, the x and y directions coincide with the h and v directions. The hardware carries out the multiplication

$$\Delta h = i_h \cdot \Delta x + j_h \cdot \Delta y + k_h \cdot \Delta z \text{ and } \Delta v = i_v \cdot \Delta x + j_v \cdot \Delta y + k_v \cdot \Delta z$$

to determine the length and direction of the line seen on the scope. Vector components are limited in length to ten bits, which corresponds to the width of the scope field.

The starting point of a sequence of vectors is a setpoint (h,v) preceding that sequence in the display list. All vectors are plotted from the end of the previous vector or a setpoint. An example of a display list and its corresponding picture, as shown in Figure (3), is helpful in understanding this idea. Note that vector 3 has a z component associated with it but that Δz is not represented graphically as long as the matrix is unity. As the k_h and k_v elements are changed from 0, the Δz_3 will begin to show as a projection on the scope. Setpoints, rotation matrices and line commands may be sequenced in any order to achieve various kinds of pictures. Two examples in Figure (4) demonstrate quite different uses of these commands.

Supervisor Program for ESL Console

In order to permit the time sharing user to use the ESL Console, a supervisor program resides in A Core of the 7094 Computer and provides subroutines which can be called to alter the display list and cause certain real time operations on the display list. This program is called DSCOPE and is described in detail in Reference 2. Three main features of the program are the attention stack, the real time operations and the display list transfers.

[illegible]

5	i_h	j_h	k_h	Rotation Matrix
6	i_v	j_v	k_v	
3	H1		V_1	Setpoint
0	ΔX_1		ΔY_1	Vector 1
0	ΔX_2		ΔY_2	Vector 2
1	ΔZ_3			Vector 3
0	ΔX_3		ΔY_3	
3	77777	3		End of File

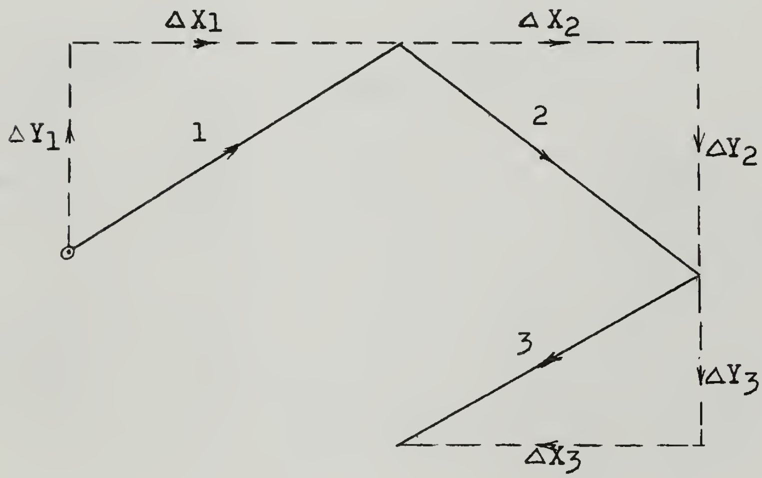


Figure (3)

Scope presentation with rotation matrix set to unity.

Rotation Matrix
Setpoint 1
Vector 11
Vector 12
Setpoint 2
Vector 21
Vector 22

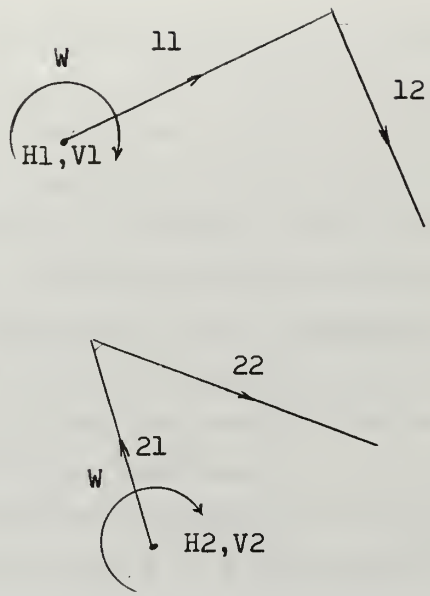


Figure (4a)

Rotation about two setpoints controlled by one matrix.

Rotation Matrix
Setpoint
Vector 11
Vector 12
Rotation Matrix
Vector 21
Vector 22

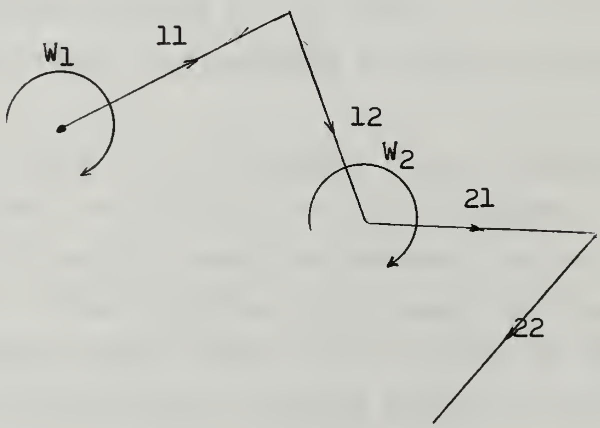


Figure (4b)

W_1 is rotation about the setpoint, W_2 is the rotation about the end of vector 12 causing compound rotation about the setpoint.

The attention stack stores appropriate information whenever an interrupt occurs in reading the display list. Interrupts are caused by pushing a button, completing a "rubber band line" or by the lightpen seeing a line on the scope. The program notifies the user of the kind of attention generated by displaying on the scope a B for button, L for line and P for the pen seeing a line. When the attention has been processed by the user's B Core program, the letter disappears.

Real time operations are routines which interpret the movement of the knobs, globe or lightpen to cause the display list to be altered as described by the following real time functions:

1) Rotation about any of the scope coordinate directions,
2) Translation of a setpoint in the h or v directions,
3) Expansion of the scale by successively doubling the dot spacing of the displayed vectors,
4) Reduction of the scale by uniformly reducing the magnitude of all the elements of the rotation matrix,
5) Making the setpoint coincide with the position of the pen track so that the picture follows the pen track.

Appendix A describes how these real time functions are used in the drawing system.

Transfer words may be inserted into the display list to make it possible for the console to read the display list in any desired sequence. This facility is very helpful when the display list is subject to many modifications. These transfers also permit the console to read the same set of commands several times in one cycling of the display. Transfer words are indicated by a 3 in the prefix and tag.

B Core System for ESL Console

The DSCOPE program may be used directly by a programmer or indirectly through a set of library subroutines that organizes the display list in a manner that makes it easier to manipulate. This set of subroutines is the B Core System for the ESL Display. It is described in detail in Reference 3.

The B Core System organizes everything plotted on the scope as a sequence of objects strung together in a pointer string. Each object has a name which may or may not be assigned by the user. This name points to a register called the display register, in which is stored the location of the beginning of the object in the display list and a pointer to the display register of the next object in sequence. The B Core System subroutines used in this report are summarized in Appendix A.

III. DESCRIPTION OF DRAWING SYSTEM

General Description

The purpose of this system is to enable the ship designer to build three-dimensional models of a ship's arrangement in the computer and display them on the face of the ESL Display in order to quickly determine the best arrangement of equipment and functions. The drafting system can be thought of as an 80" square drafting board upon which can be constructed three-dimensional wire figures of a ship and its contents. The board can only be viewed through a 10" window (the scope) but the drawing board may be moved behind the window and may be temporarily shrunk so that more of it may be looked at through the window. The board can be rotated about any point located in the plane of the board. The designer may set the scale of the figures drawn on the board to any of 7 reductions ranging from 64' = 1" to 1' = 1". Each change of scale by 1 step results in a doubling of the scale.

At present only crude drawings can be constructed out of straight line segments but other programmers have recently worked out more accurate three-dimensional drawing capability.[4]

The primary facility of this arrangement system is the ability to call objects out of a file and push them around on the board, rotate them in the plane of the board and raise them off the board as desired. Objects which do not exist in the file may be constructed using the crude drawing feature. Objects drawn may be treated in the same manner as objects called from the file. The arrangement obtained by assembling objects may be modified by addition, deletion or rearranging as often as desired while the picture remains on the screen. The present program does not permit filing the arrangement in such a way that it can be called onto the screen again for more work. Once an arrangement has been completed, it can be filed but not altered. Further discussion of this shortcoming is contained in Section V under Working Pictures.

---

[4] The subroutines for this facility have been written by Robert Polansky of the Electronic Systems Laboratory. A complete report is expected as a MAC memorandum in June 1965. These routines are of a general enough nature to be applied to a variety of uses.

Properties of scale, weight and name can be assigned to each picture and are filed with the picture. Scale is assigned automatically according to the scale in which the object was drawn or arranged. Weight and name are assigned by the designer via the teletype.

Functions of Buttons and Knobs

Each of the controls used by this program has certain functions associated with it, which enable the designer to develop an arrangement plan and evaluate it. Some of these functions change according to buttons pushed by the designer. The functions of the knobs switches and pen are included with this summary of button functions.

Button

s - PENTRACK - Initiates the pen tracking cross the next time the pen sees light on the scope.

1 - LINE START - Initiates a "rubber band line" at the pen track position. Any movement of the pentrack causes a line to stretch from the initial position to the new position of the tracking cross. If a "rubber band line" is already active, this button completes that line as well and leaves a line attention in the attention buffer.

2 - LINE FINISH - Terminates a "rubber band line" and detaches the tracking cross from the line.

3 - RESTART - Not assigned to any special purpose but is useful for restarting the picture if the display gets hung up and does not exhibit the lines in the display list.

Note: These first 4 buttons are all assigned by the DSCOPE program. The remainder of the button functions have been assigned by the ship arrangement system of this report.

4 - HORIZONTAL VERTICAL - Sets a mode "T" which constrains all "rubber band lines" to become vertical or horizontal when they are processed. To indicate that the system is in this mode, a "T" is displayed in the lower left corner of the screen. Pushing the button again removes this mode.

Note: If an attempt is made to draw a picture on the screen before a setpoint has been plotted, the system will print an error comment telling the user to push button 9.

5 - SIGN OFF - Signs off the scope. Removes all pictures and the program goes to dormant. The program may be restarted from scratch by typing "start" on the teletype.

6 - NAME - Causes the first 10 characters typed on the teletype to be displayed on the scope immediately following the last line of the display. Button 11 may be used in conjunction with this button to set or unset a mode "F". In mode "F", the letter "F" is displayed in the lower left hand corner of the scope and only the display of characters as described above occurs. If the mode is not "F", the following also occurs. The first 5 letters displayed become the name of the last picture on the screen and the picture is completed. Completing means that the aggregate of lines drawn since the last complete picture become identified as one picture, which may be moved around as a unit. No more "rubber band lines" may be added to a complete picture. Complete pictures are discussed more fully in Section IV.

7 - FILE - After the button is pushed the first picture seen by the lightpen is filed in a disc file. If the picture has a name, it will be printed on the teletype, otherwise the name of the picture will be requested by the routine.

8 - LOAD - Copies an object out of the "Ship Parts" file onto the screen with the x, y directions of the picture coincident with the h, v directions of the scope. The origin of the object is the position of the light pen tracking cross when the button is pushed. The real time functions of rotation and translation are set to operate on this picture via the globe and knobs. The name, weight and scale information originally filed with the picture is retained. The picture is plotted in the present scale of the arrangement as currently indicated on the two rightmost decimal switches.

9 - COMPLETE - Completes a picture without naming it or plots a setpoint and rotation matrix for the start of a new picture at the position of the pen tracking cross.

10 - MATRIX DEBUG - Resets the currently active matrix and rotation buffer to unity. This button is necessary to correct for a shortcoming in the hardware which causes the rotation matrix to go to zero unexpectedly sometimes.

11 - FINISH - Turns the "F" mode on and off.

12 - COMBINE - Attaches completed pictures together and puts them under the control of the setpoint and rotation matrix of the first picture plotted. After this is first done, the first picture plotted becomes the reference for all other pictures plotted during the same session at the console. This first picture essentially represents the drawing board mentioned at the beginning of this section. Pictures which are not completed cannot be attached. Attaching may be done as often as desired. The attaching is done by recomputing the x and y components of the complete pictures according to the value of the respective rotation matrices and running a blank line from the end of one picture to the beginning of the next. During this operation the x and y beginnings and endings relative to the first picture are saved as properties of each complete picture. The z information does not get altered until button 15 is used.

13 - MOVE PICTURE - This routine allows the designer to point to any picture with the lightpen and gain control over it for translation and rotation. This is done very simply if the picture seen by the light pen already has a setpoint and rotation matrix. If the picture has been attached to other pictures by button 12, it will be temporarily removed from the chain of attached pictures and given a setpoint and rotation matrix. To indicate that this special condition exists, an "M" is displayed in the lower left hand corner of the scope. There are two ways this loose picture can be re-inserted into the chain of attached pictures. One way is by pushing button 12, which will not only reattach the picture, but attach any other complete pictures that remain unattached.

The other way is by pushing button 13 again. When performing this latter function, button 13 will ignore any pictures seen by the lightpen and will only reattach the loose picture and return translation and rotation control to the picture that was controlled prior to pulling the picture out of the chain. If some other picture is to be controlled, button 13 must be pushed again.

14 - CONSOLIDATE - This routine combines all the attached pictures into a single complete picture, FP. Completing the arrangement in this manner makes it possible to file the arrangement as a single unit. No modifications except addition can be made to the arrangement completed in this manner.

15 - MOVE Z - Permits the designer to move pictures along the Z axis of the reference picture. Pictures must be attached or an error comment is printed "wrong picture seen". The procedure is 1) push the button and point to the picture that is to be moved; 2) rotate the reference picture into the x, z plane, position the pen track and push the button again; 3) move the pen track the vertical distance that the picture is to be moved and push the button again. The picture will then be moved along the z axis by the amount and direction generated by the vertical distance between the two successive pen positions. The z beginning of the picture relative to the reference picture is saved.

16 - DRAG - Permits an alternate method of moving an object around on the drawing board by dragging the picture with the lightpen track. This subroutine assumes that the reference is in the xy plane so inaccuracies will occur if the picture is in any other plane. The procedure is to position the pen track, push the button and then point to the picture to be dragged. On completion of this routine the picture will follow the pen track. If any rotation of this picture is desired, it must be assigned as usual by button 13. To stop dragging, button 16 must be pushed again.

17 - ZOOM - Pushing this button enables the right knob to be used to collapse the reference picture plus all attached pictures

down in scale and sets the globe to control rotation of the reference picture if this is not already the case. No other operation can be performed on the picture while in this mode. Pushing the button again returns control to the condition existing prior to the first push.

18 - SCALE CHANGE - When a more permanent scale change is desired the new scale is set on the two right most decimal switches. The number on these switches represents the number of feet of full scale per inch of drawing. Allowable scales are 64, 32, 16, 8, 4, 2, and 1.

19 - CHANGE CENTER - By use of this button the designer can change the point about which the reference drawing can be rotated. This permits the drawing board to be moved behind the window of the scope field for work anywhere on the drawing board. The procedure is: first place the pen track at the point about which the reference picture is to rotate, then, with the translation knobs, locate the reference picture in the desired position with respect to the pen track and push the button.

20 - ASSIGN WEIGHT - Pushing this button starts a procedure for assigning a weight to each completed picture which does not already have a weight assigned. When all pictures have been assigned weights, the designer may look at and change any of the weights. In this latter procedure the name and weight of the picture pointed to by the lightpen will be printed followed by "Name =". If the response to "Name =" is a decimal pointed number, the weight will be changed to that number. A carriage return alone leaves the weight unchanged. The designer may continue this process as long as he continues pointing at pictures.

21 - CENTER OF GRAVITY - This button causes the computer to take the x,y and z positions of all pictures relative to the reference picture and the weights and compute the total weight and center of gravity. The center of gravity will be displayed and the total weight and relative x,y and z position of the center of gravity will be printed on the teletype.

IV. PROGRAM DESCRIPTION

Programming Language Used

The routines for ship arrangement have been written in M.I.T.'s computer-aided design language, AED - 0, which is an extension of ALGOL. Instructions for the use of the language are contained in References 5 and 6. This language is especially suited for the manipulation of blocks of registers and the manipulation of bits within registers. It is easier to write than the machine language but does not produce programs which are as efficient as a good programmer can write in the machine language. Since M.I.T. is installing a new computer in the near future, it was sensible to use a language which would maintain its usefulness with the installation of the new computer. When the new computer is installed only minor changes to the programs and some of the subroutines will be required. The language also offers the advantage that the logic is easier to follow than it would be if machine language were used.

Organization of Pictures

The B Core System of Charles Lang[5] imposed a kind of order on the way in which pictures displayed were to be organized. This convention requires that each object plotted on the scope have a name. This name points to the display register for the object and can also point to other properties of the object such as scale, weight and alphabetic name or title. There are five classes of pictures used to organize the scope presentation.

1) Control objects: i.e., setpoints, SP, and rotations matrices, RM
2) Temporary pictures, TP
3) Complete pictures, FP
4) Working pictures
5) Miscellaneous objects

The control objects are the key to the ability to dynamically control a picture. Setpoint names and rotation matrix names are organized

[5] See Reference 3.

in linear arrays all indexed on J, the same index as the complete pictures. The display registers for setpoints and rotations matrices are the arrays P and M.

Temporary pictures are named TP(I) with the index I taking on values from 1 to 20. Temporary picture names are assigned to lines drawn by buttons 1, 2, and 4. The display registers for temporary pictures are in array T. The setpoint and rotation matrix for a set of temporary pictures are the SP(J) and RM(J) for the complete picture that these temporary pictures will become through use of button 6 or 9. No operations may be performed on temporary pictures after they are drawn except for rotation and translation via the matrix and setpoint, and completion into a complete picture.

Complete pictures have names, FP(J), and result from the renaming of temporary pictures or from plotting directly from the disc file. The FP's are the basic building block of the display and keep their identity through all manipulations except for completion by button 14 into FP(0). FP(0) is always the reference picture for the display being worked on; its setpoint and rotation matrix are used to control the set of FP's attached together by buttons 12 and 13. The array, F, is used for the display registers and for the storage of the properties of the individual complete pictures. The format of an array segment for one complete picture is illustrated in Figure (5).

Working pictures do not presently exist in the program but their purpose is to permit the naming of a set of complete pictures, FP's, as a single picture without the FP's losing their identity. This type of picture is necessary in order to be able to file a picture which still requires alteration and to be able to edit a displayed picture to reduce detail and shorten the display list. Working pictures are discussed more fully in Section V.

Miscellaneous objects include the indicator letters and aids to the draftsman that are not associated with the organization of the picture sequence. Some examples of miscellaneous object names are NOF, TEM, CG, REFR.

[illegible]

Display Register			
	HE		VE
	XX		YY
WEIGHT			
	SCALD	ISV	SVPIC
NAME			
	ZZ	IWT	SVSPT
ZE			

HE,VE,ZE - end point of picture.

XX,YY,ZZ - beginning point of picture.

WEIGHT - assigned weight of picture.

SCALD - scale picture is drawn in.

SVSPT - points to saved setpoint for this picture.

SVPIC - points to the saved picture.

ISV - indicates whether SVPIC and SVSPT being used.

IWT - indicates whether weight has been assigned.

Figure (5)

Format of F array for a single picture.

Definition of Variables.

Certain variables have special meanings associated with them. Some are used in common throughout the program while others are only used locally.

XO, YO are the present position of the reference setpoint relative to the center of the scope. Whenever these values are computed, the RM(O) matrix is assumed to be unity.

H1, V1 are usually used for chasing through the vectors in a picture to compute the h, v position of the end of a set of vectors.

H2, V2 is usually used for temporary storage of the value of a setpoint.

HSAVE, VSAVE saves the position of the end of the last attached picture relative to XO, YO.

I is the index of the temporary pictures.

J is the index of the complete pictures, rotation matrices and setpoints.

K, L, and KM are temporary indexes.

OLDR, OLDT are used to remember the name of the last rotation matrix and setpoint on which rotation and translation were effective.

OLDD remembers the name of the setpoint being dragged.

SAVEJ saves the J index of the next complete picture that will be attached the next time button 12 is pushed.

IX, JX, IY, JY are the elements of the rotation matrix.

A is an array used by the procedure ATTN(A) for storing attention information.

BOX, CURPIC, CRATE are used for temporary pointers to blocks of free storage.

ROTMAT is a boolean variable that is true when the last picture in the display list is complete and a new setpoint and rotation matrix has not been plotted.

CPADR is the relative location in the disc file "Ship Parts" of the beginning of the next picture to be filed.

CONADR is the relative location in "Ship Parts" file of the table of contents information for the next picture to be filed.

1	CONADR	CPADR
2	NAME1	
3	LNGTH1	PADDR1
4	NAME2	
5	LNGTH2	PADDR2
	etc.	
20	NAME10	
21	LNGTH10	PADDR10
22	Pictures filed here.	
1400		

Figure (6)

Format of "Ship Parts" file.

LNGTH is used for the length of the picture that is being filed on the disc or being dumped into B Core.

PADDR is the relative location of a picture in "Ship Parts" file. The format of the "Ship Parts" file is illustrated in Figure (6).

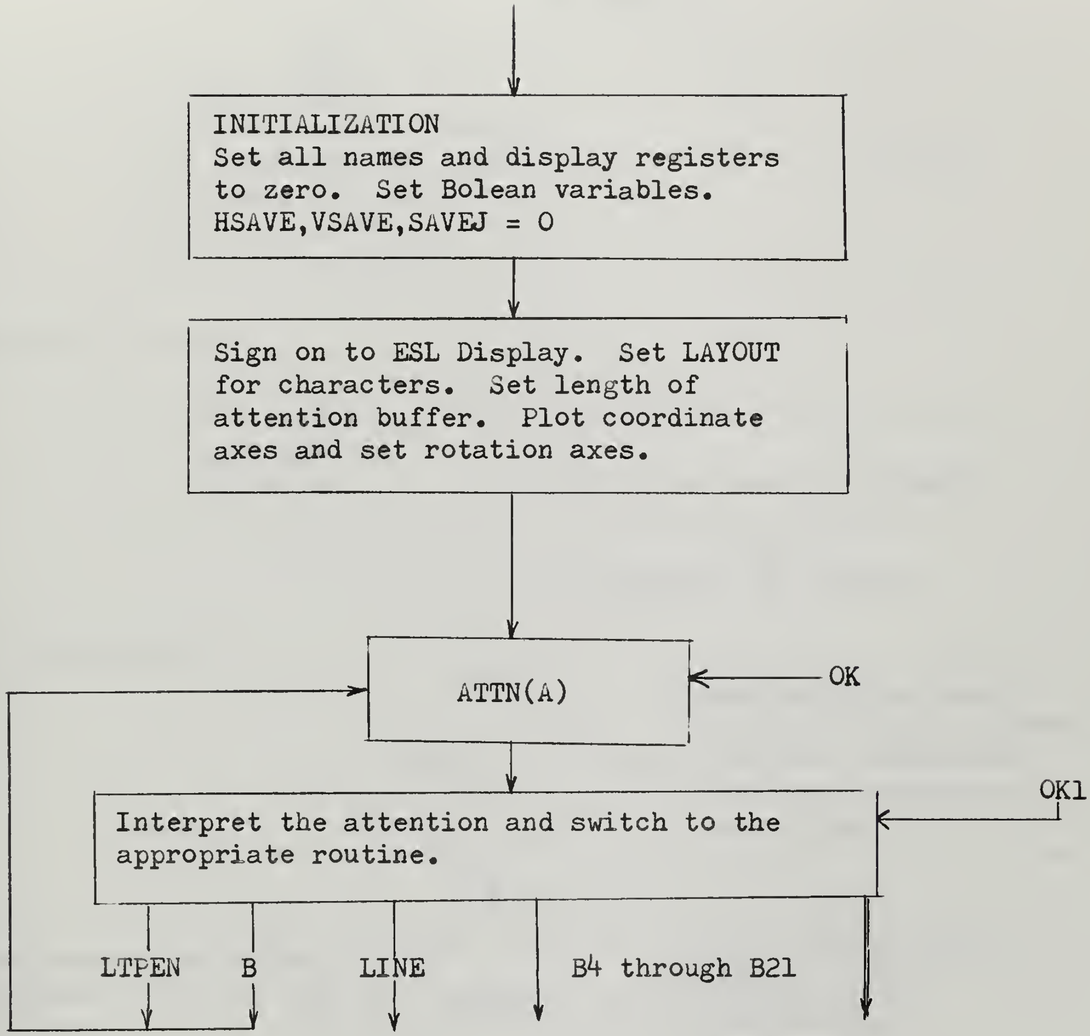
POUND is a real variable used as temporary storage for weight information.

MOMX, MOMY are real variables used in computing the center of gravity.

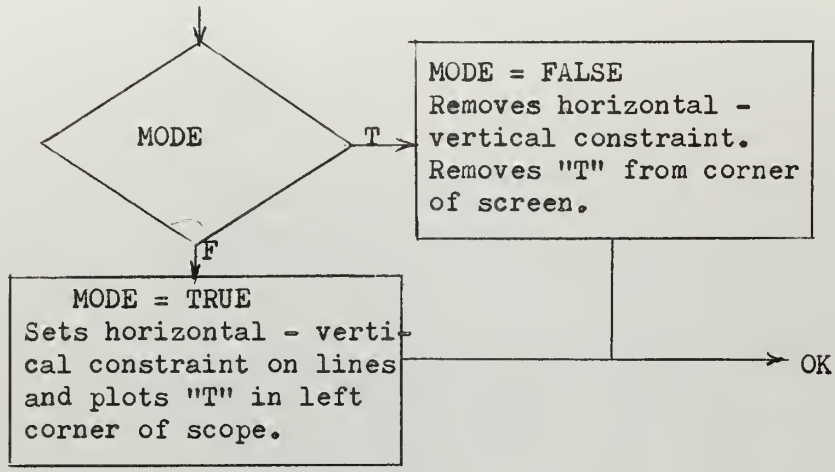
Flow Diagrams of Button Routines

The following pages are the flow charts of the computer routines for the ship arrangement system. The computer source programs are available in Reference 7. Copies may be obtained on request to the Department of Naval Architecture and Marine Engineering, Massachusetts Institute of Technology.

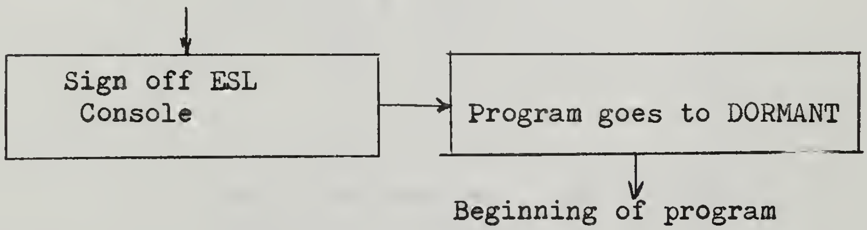
Program start



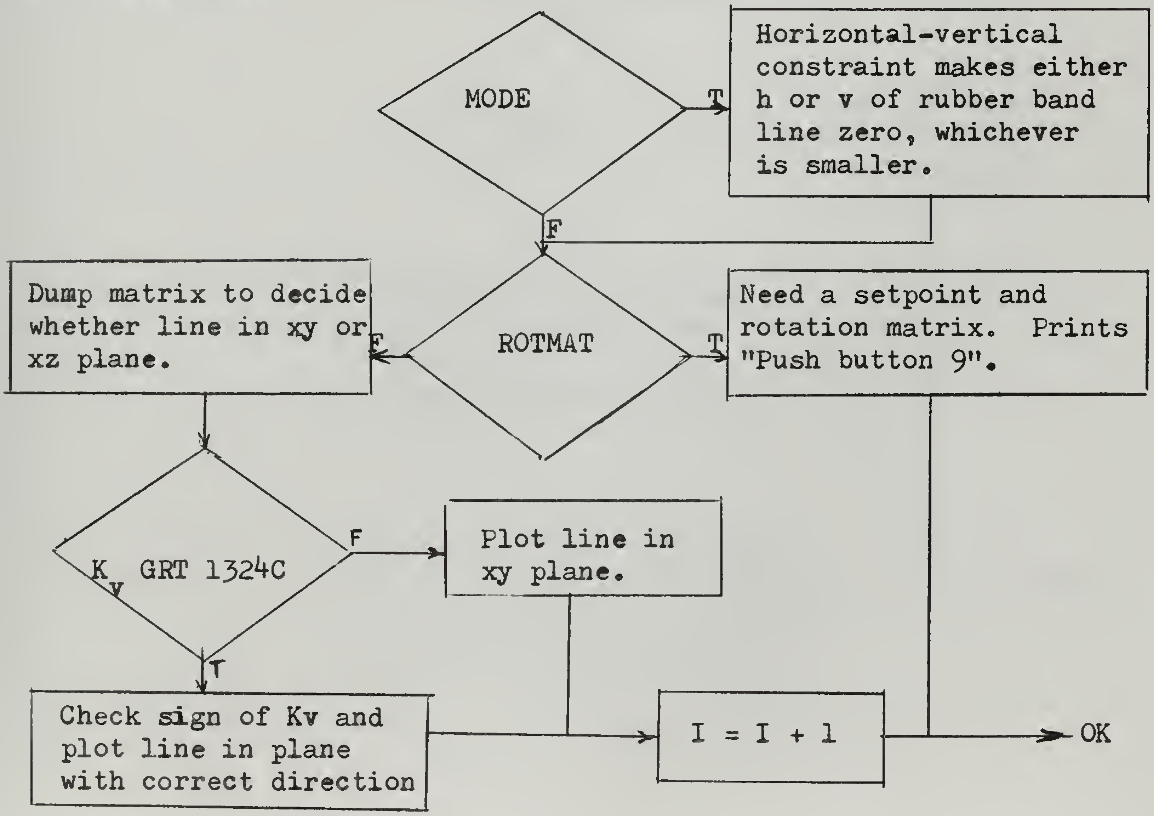
B4 Horizontal and vertical constraint mode.



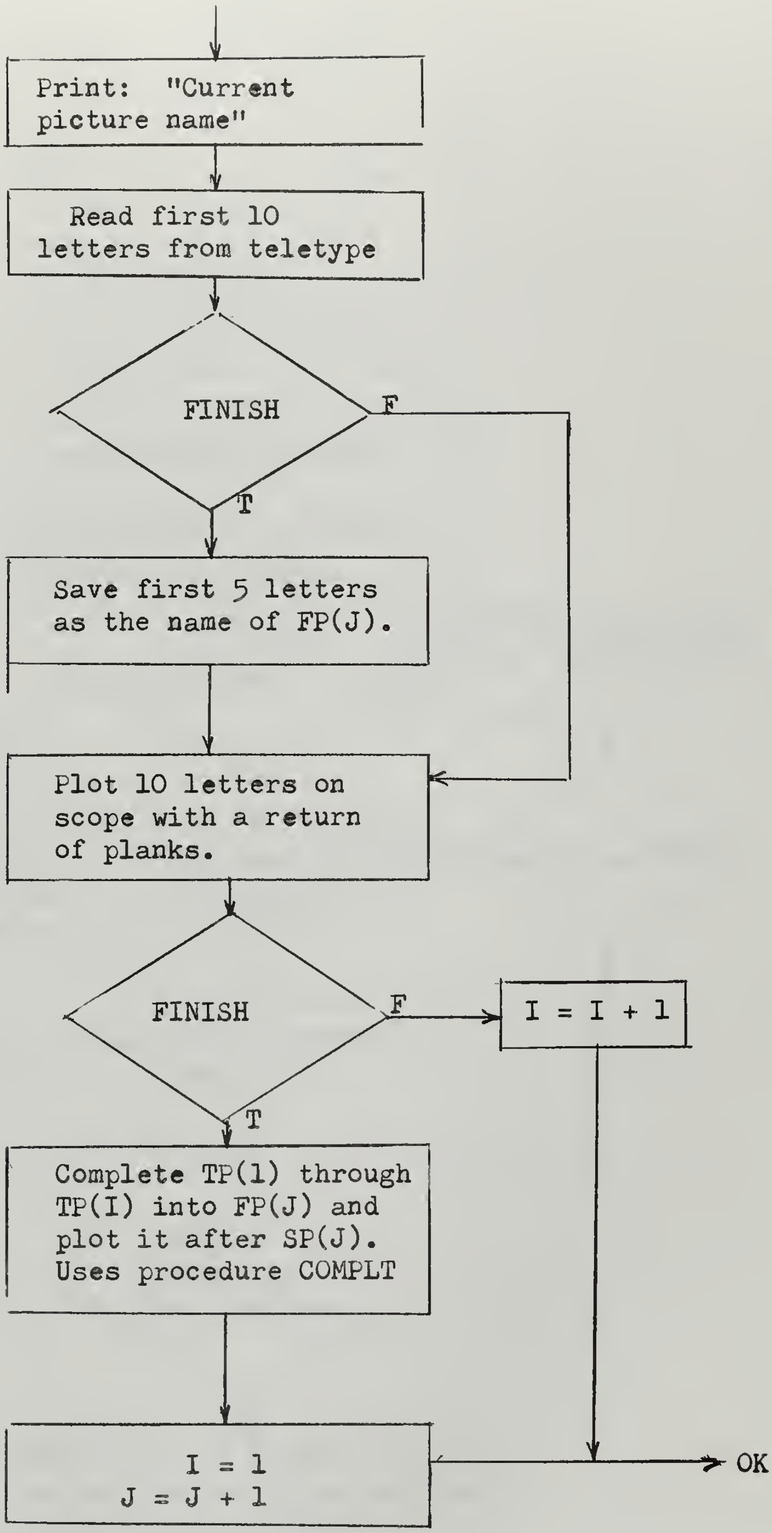
B5 Sign off console.



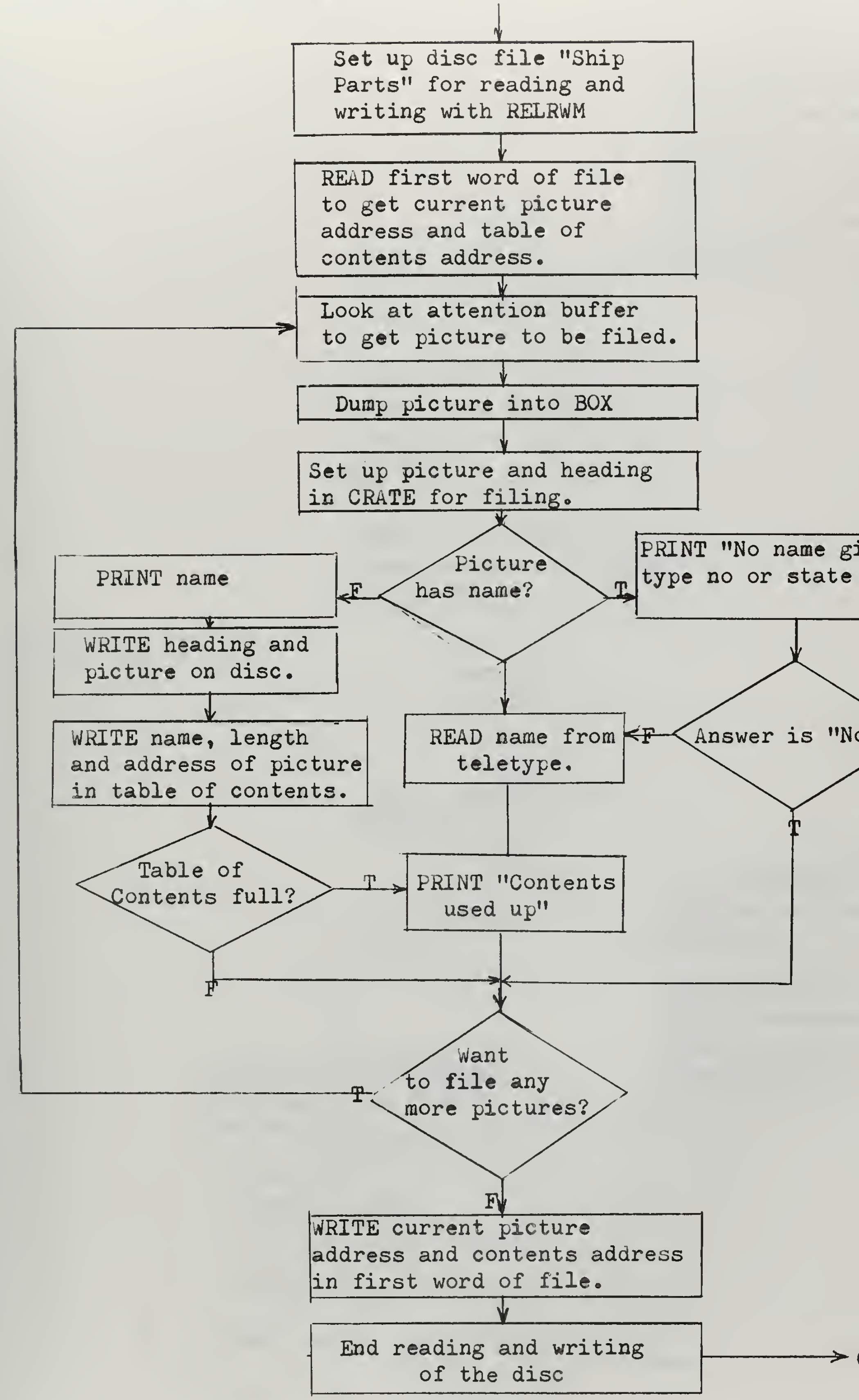
LINE Draws lines.



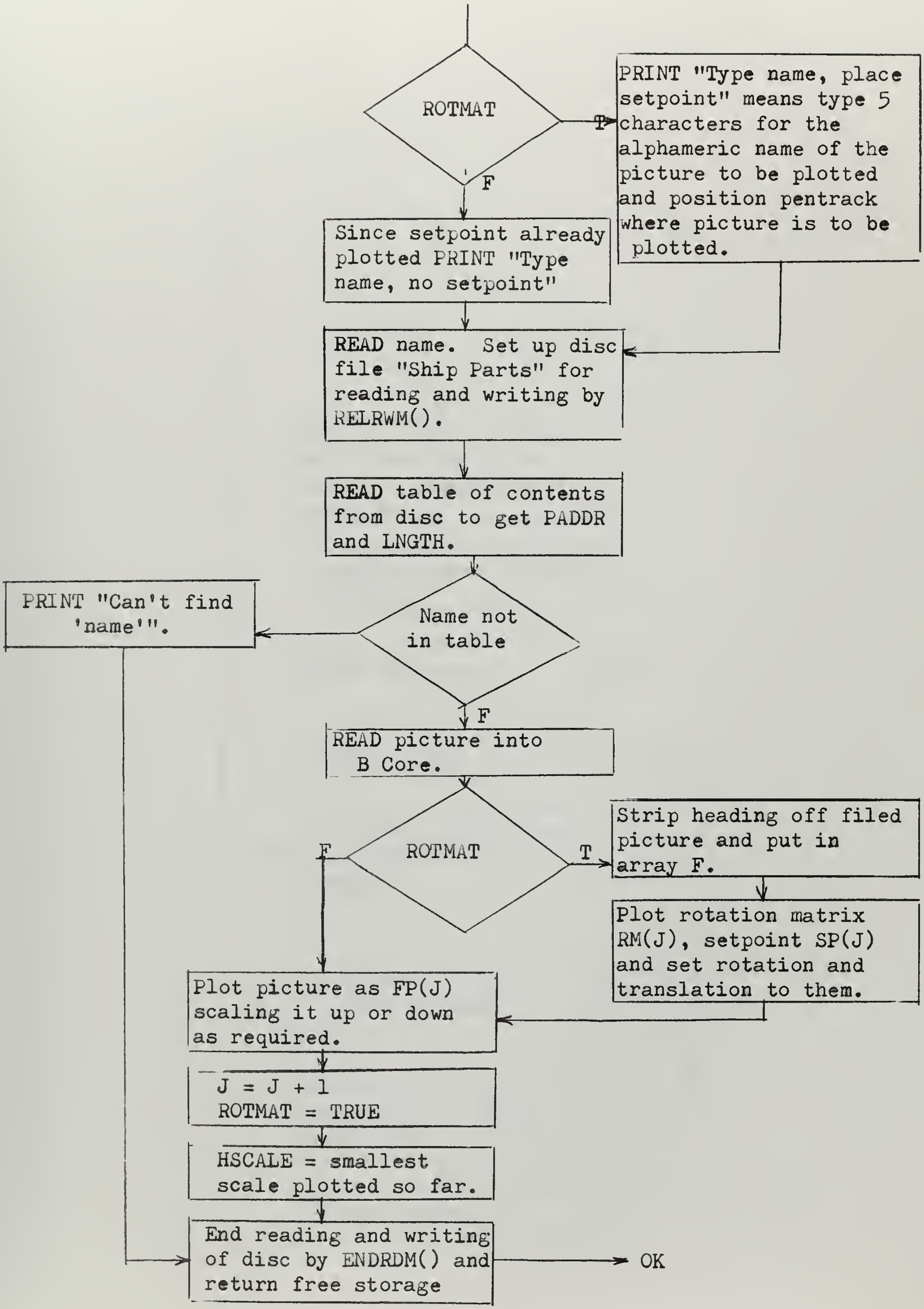
B6 Display characters



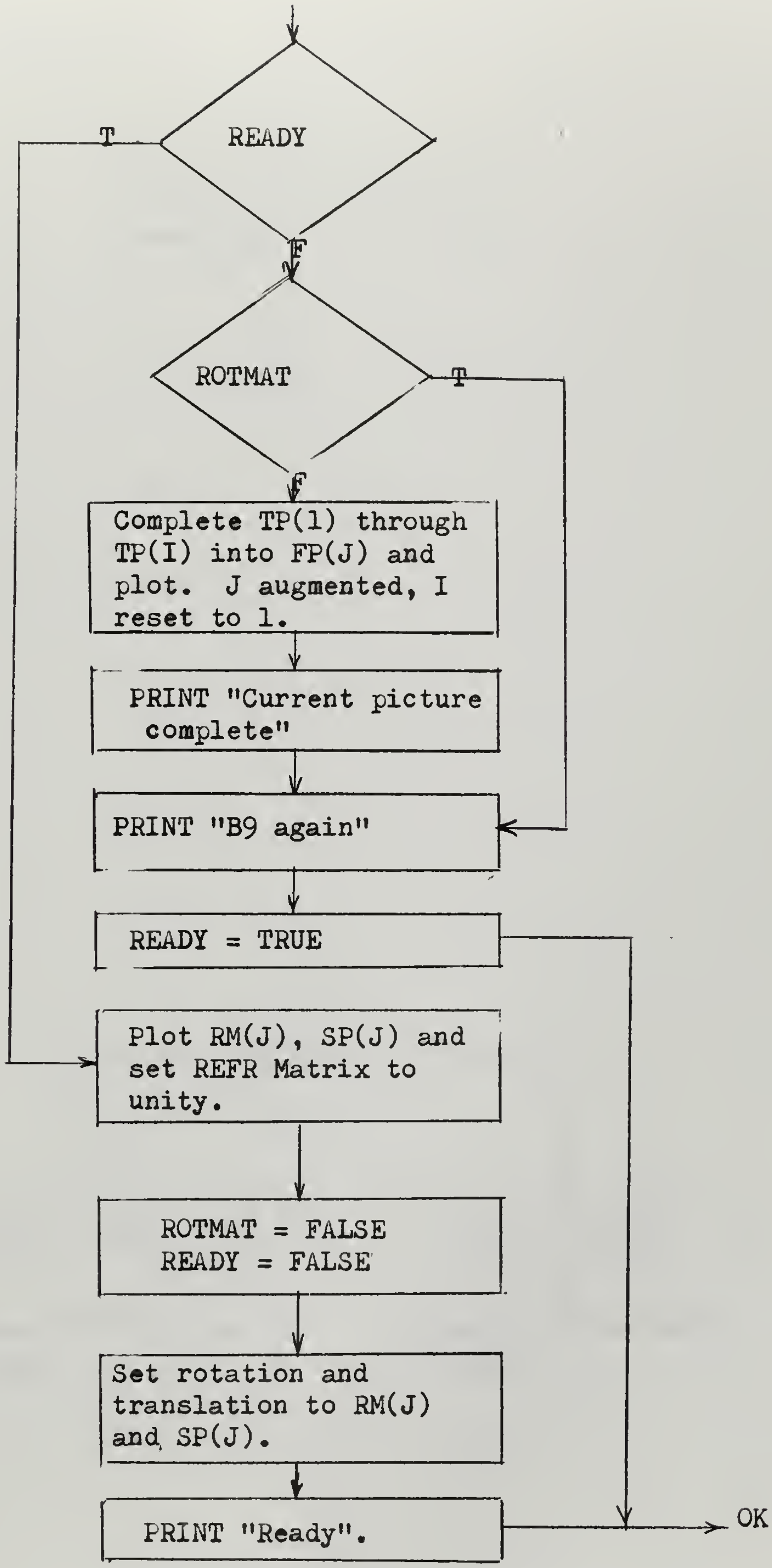
B7 FILPIC Files picture on disc



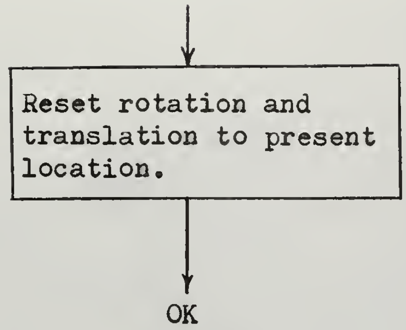
B8 LODPIC Plot picture from disc file.



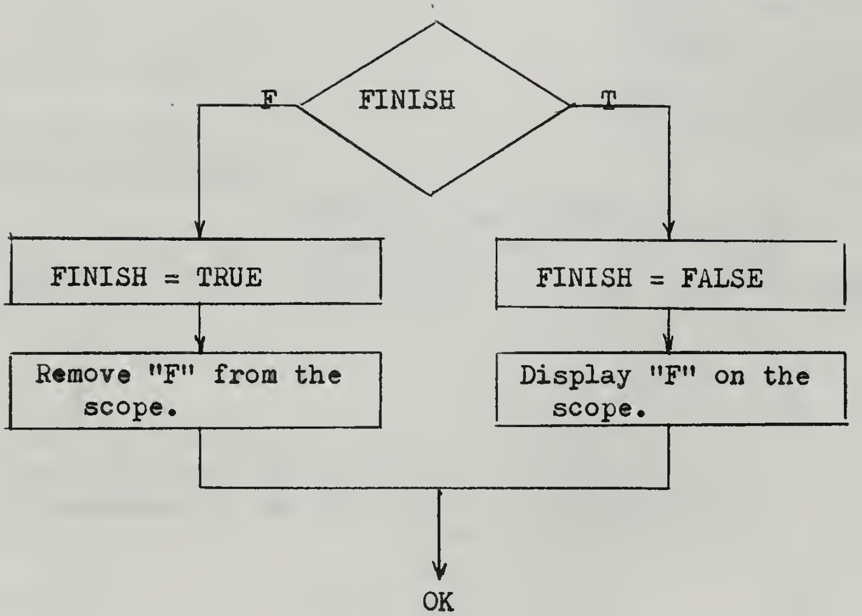
B9 Complete picture.



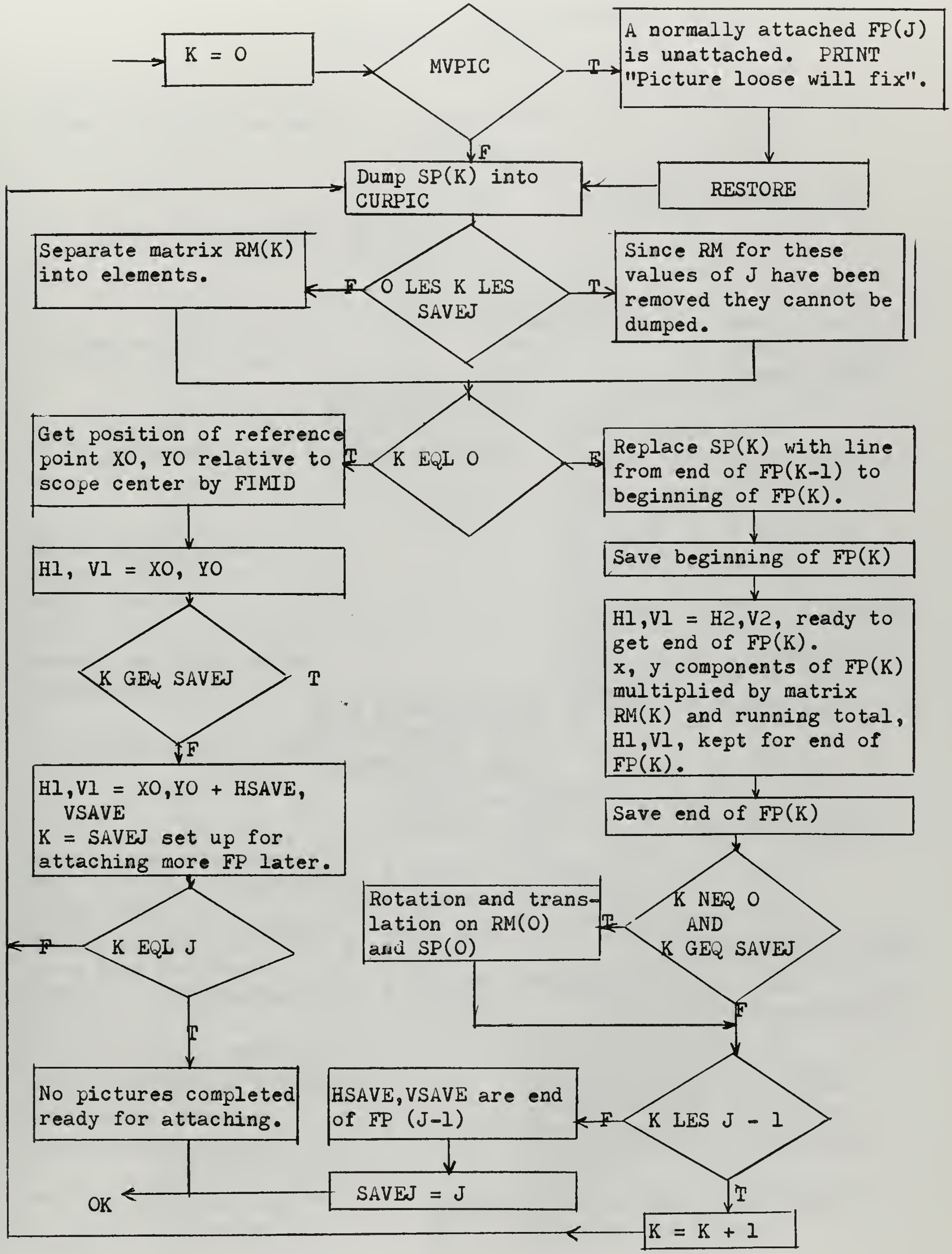
B10 Matrix Debug



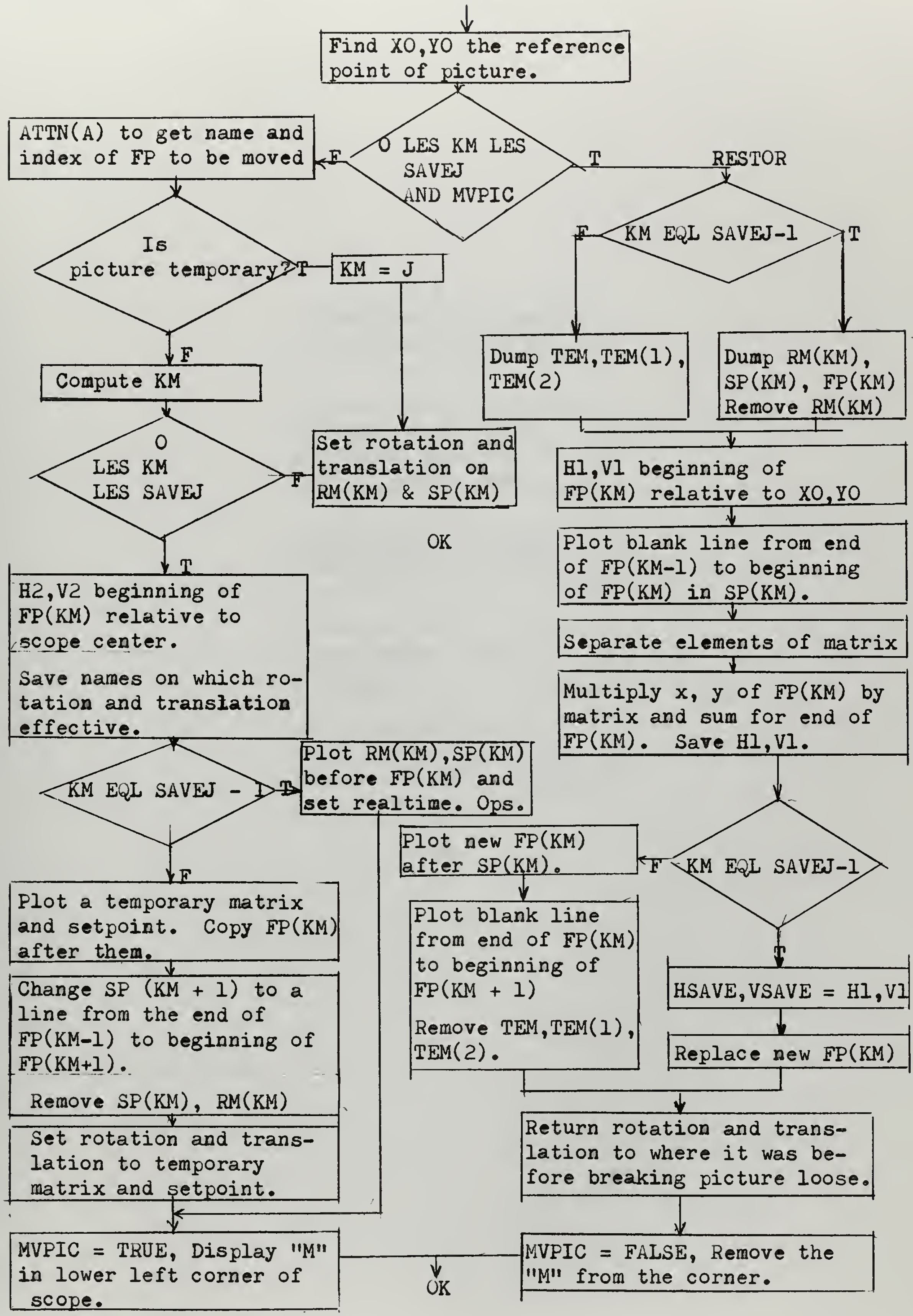
B11 Finish



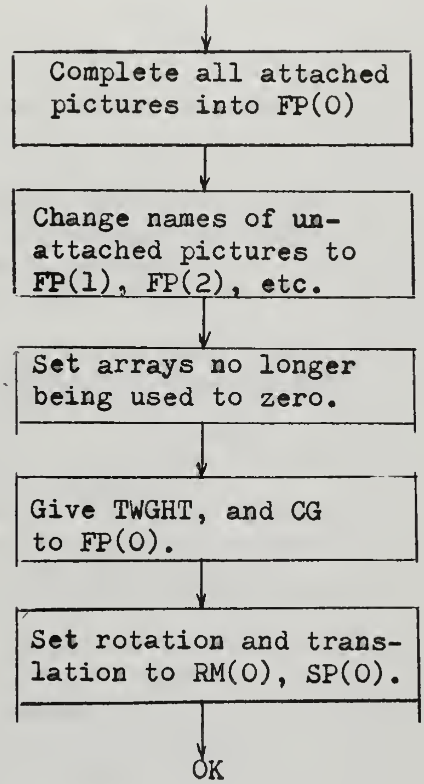
B12 COMBINE attaches pictures together.



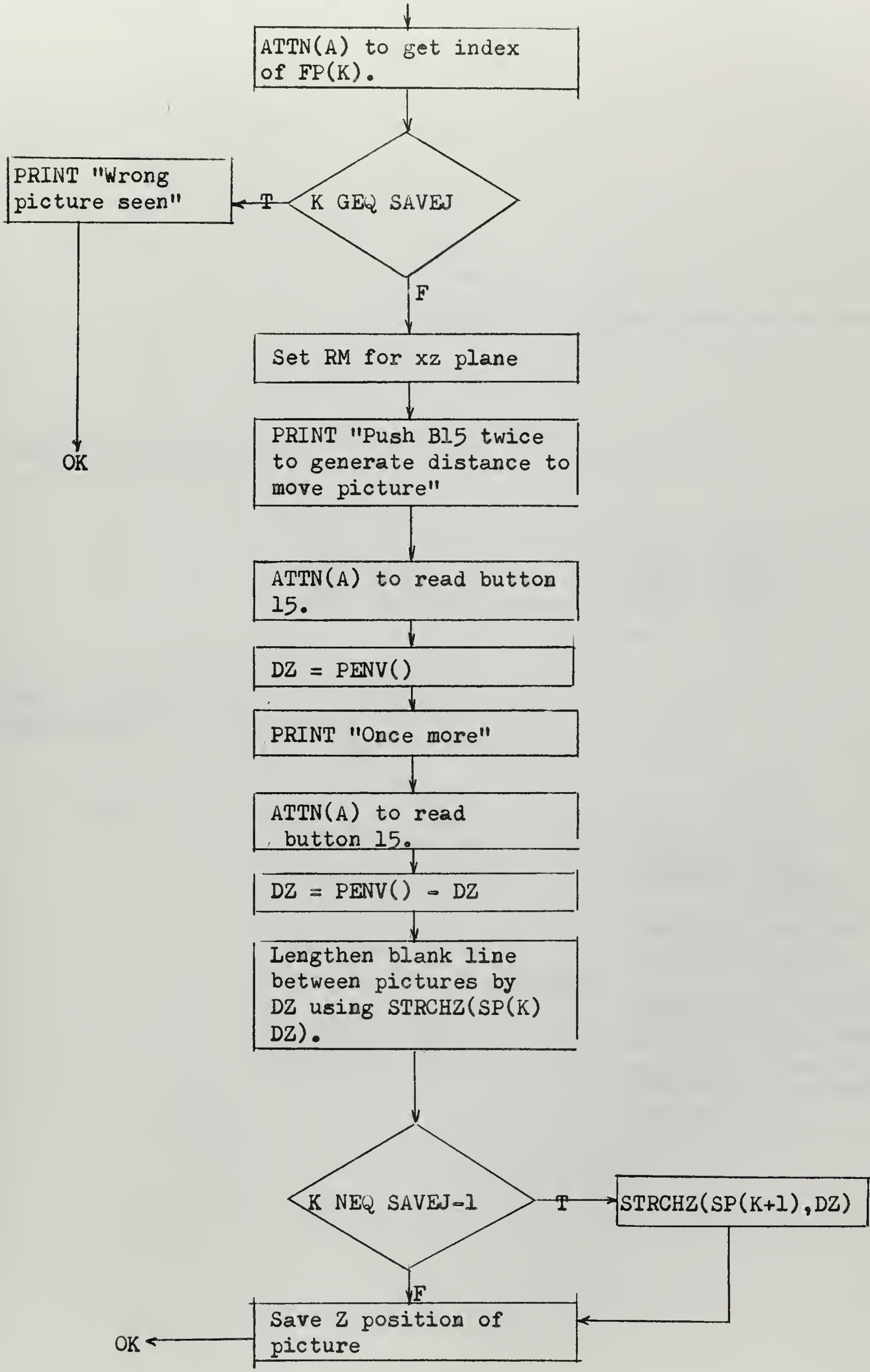
B 13 MOVPIC Rotate and Translate picture.



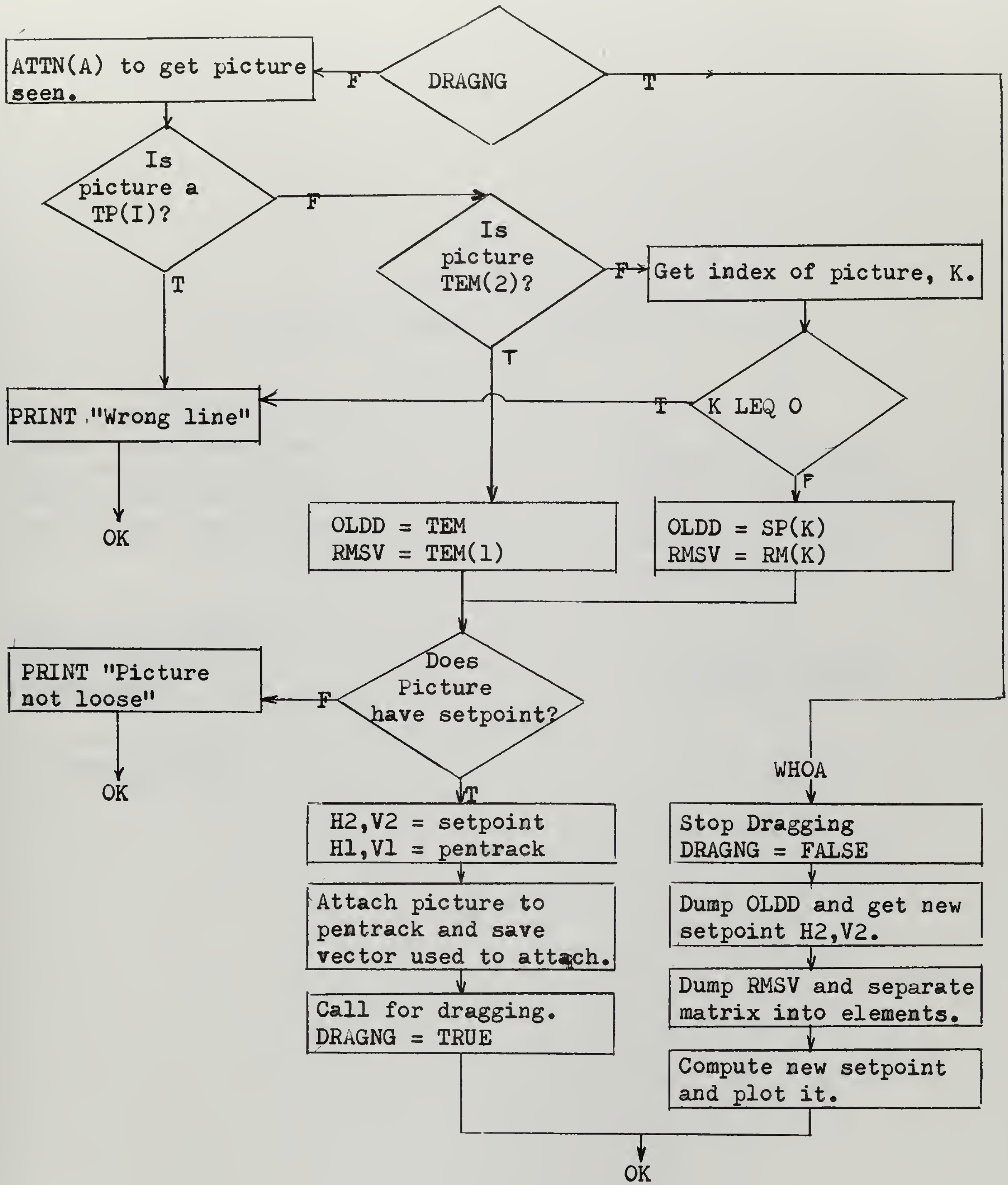
B14 COATPI Consolidate attached pictures



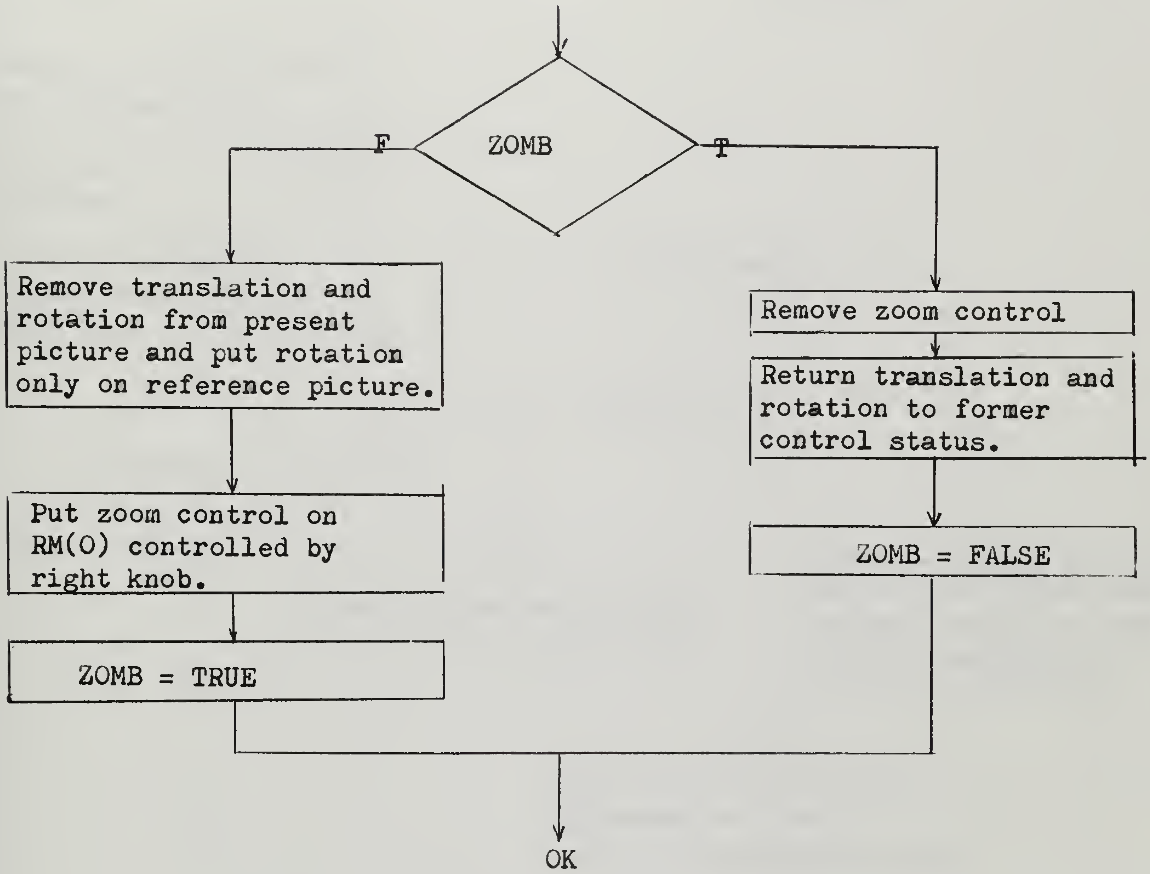
B 15 MOVES Moves picture along Z axis.



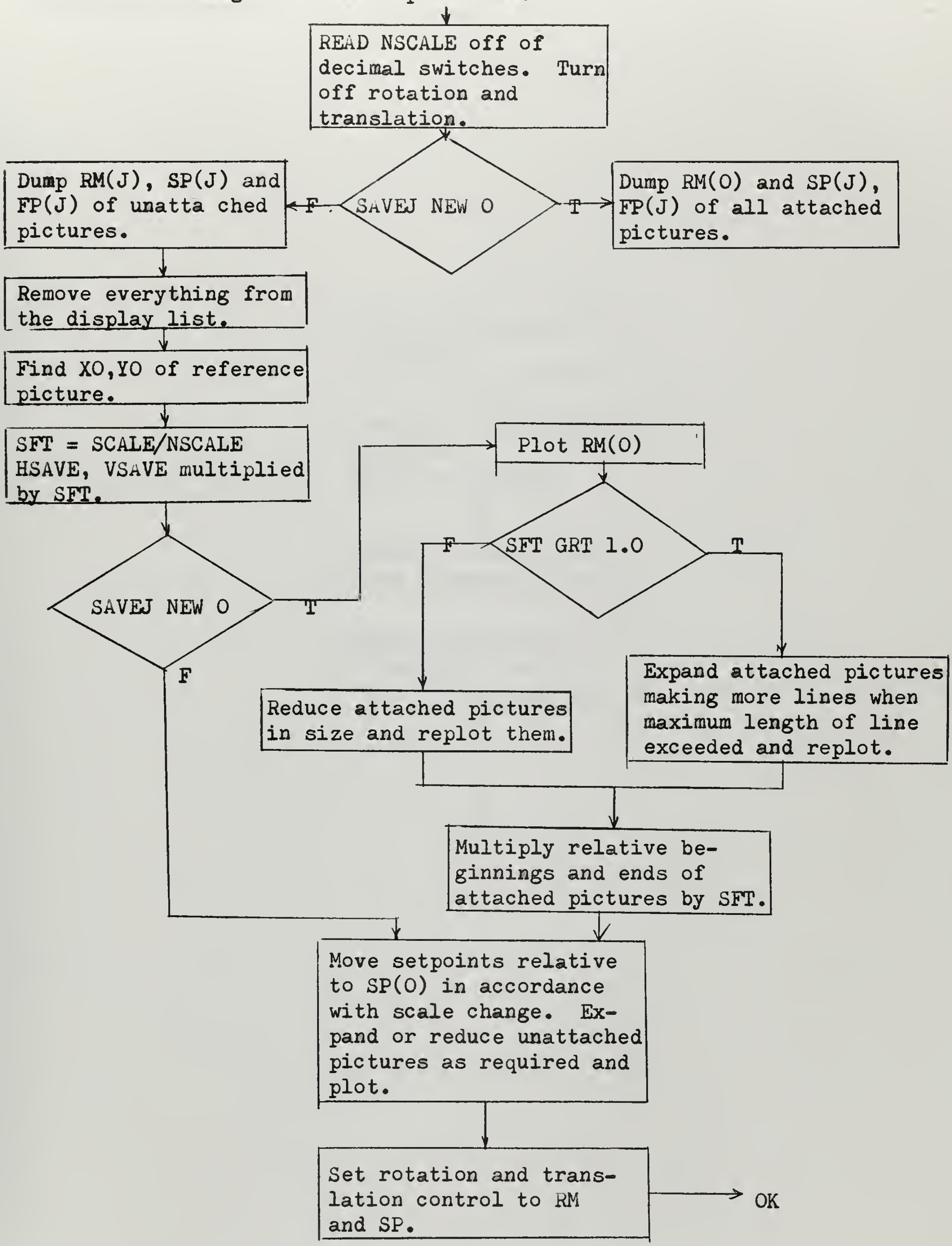
B16 DRAGR Drag a picture



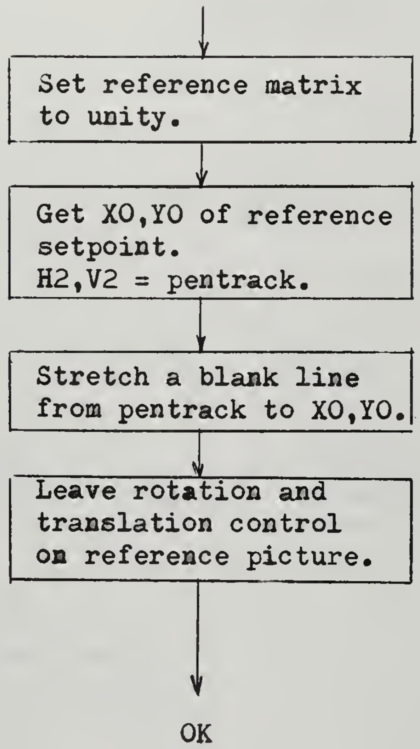
B 17 ZOOM Collapse and magnify drawing.



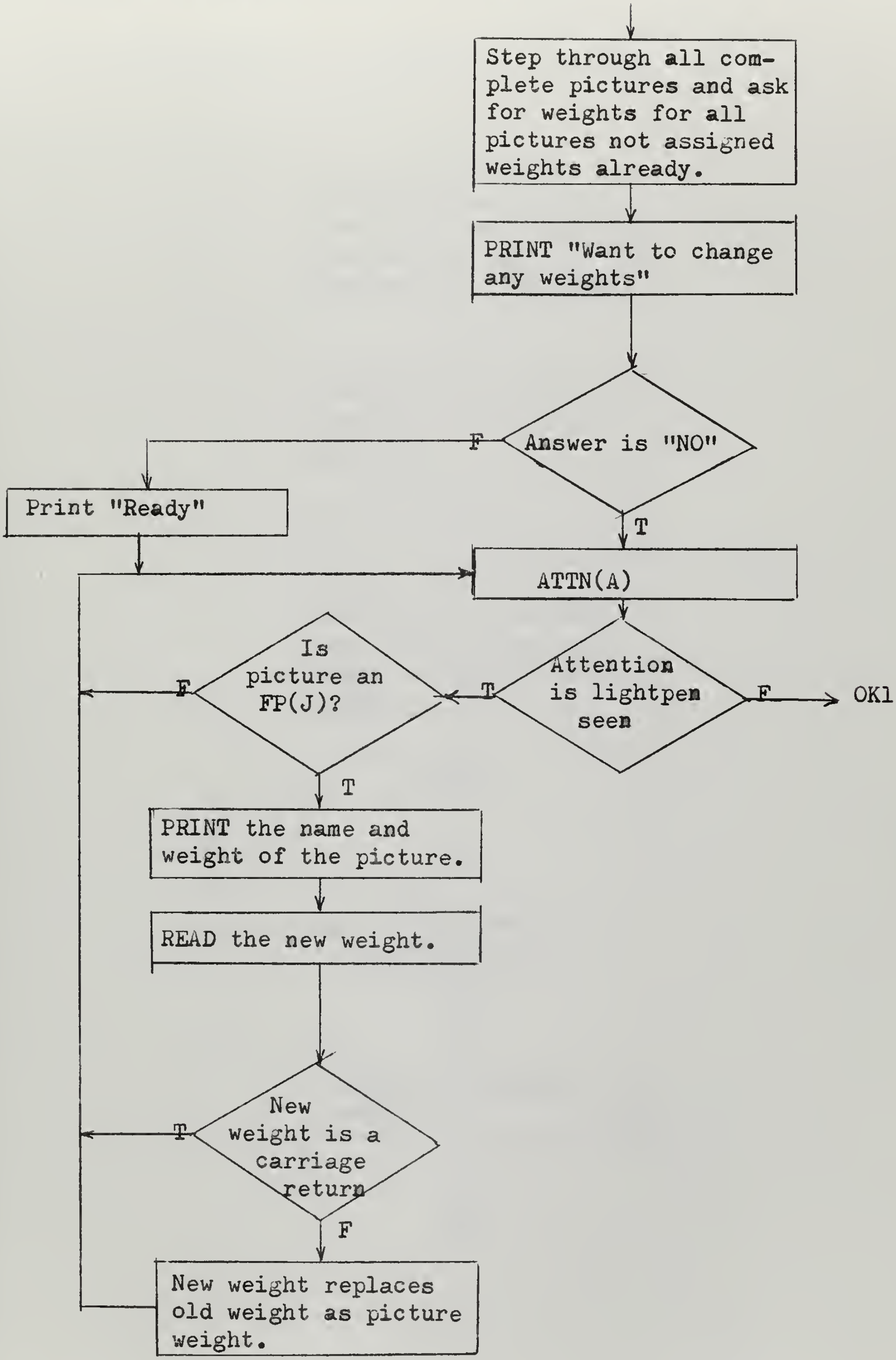
B 18 SCALVD Change the scale up or down.



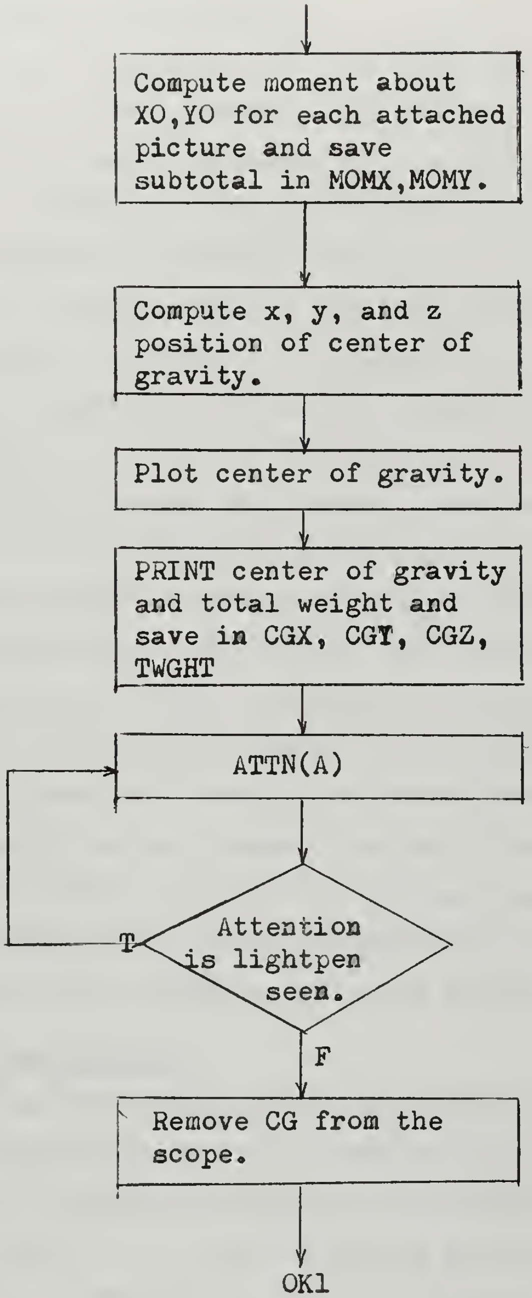
B19 CHGCTR Change center of rotation.



B20 ASGNWT Assign weight to pictures.



B21 GRAVCTR Compute the Center of Gravity.



V. DISCUSSION OF METHODS AND UNCOMPLETED WORK

Alternative Organization of Pictures

The present system organizes the pictures names in arrays. The temporary pictures, TP, are indexed on I; the complete pictures, FP, rotation matrixes, RM, and setpoints, SP, are indexed on J. All information about a picture is accessible when the index is known. Though this organization is somewhat rigid it has been sufficient for the system as so far developed. The array organization is easy to use and economical of storage and program space for the kinds of drawings within the capability of the ESL Display.

A more flexible organization of pictures is that based on pointers which thread their way through the picture names in both directions. This organization is illustrated in Figure (7). It offers two advantages. First, the maximum sequence length need not be predetermined but can be continually expanded and retracted by the free storage facility of AED. Second, removing or adding pictures from the middle of the sequence is easily accommodated. Since the temporary pictures are the vectors out of which are constructed the objects in the arrangement, the flexibility available in a pointer string is well suited for their organization. The sequence of lines in storage can always follow the sequence of the display list, making modifications of the displayed picture by actions of the lightpen and knobs fairly straight forward.

Proposal for Line Manipulation

The programs as now written do not develop the potential for drawing and modifying pictures by manipulating lines on the screen but some facility in this is necessary in the ship arrangements problem. It would be very useful to be able to take a picture already drawn, say the plan view of a waterline, and move the waterline in and out in the vicinity of a point or line segment by grabbing the line at some point with the light pen track and making the point follow the pen. This idea is illustrated in the Figure (8).

The pen track sees a line at 1, moves to 2 and has lines on either side change length as the center line is moved in the plane of the scope.

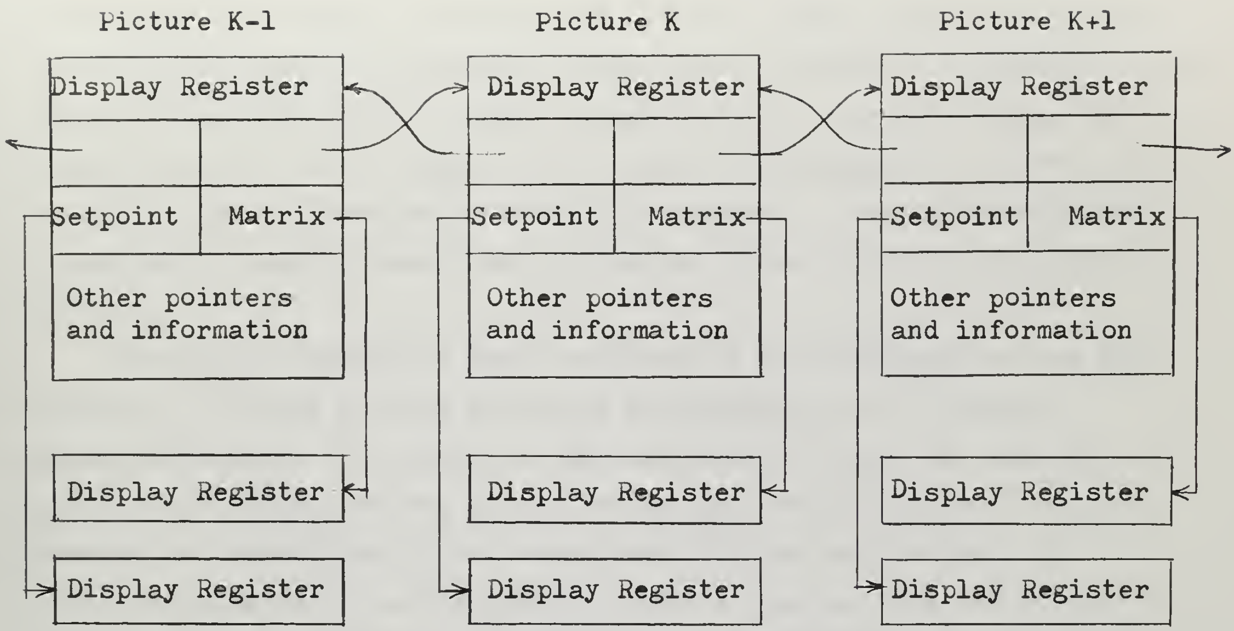


Figure (7)

Pointer string storage of pictures.

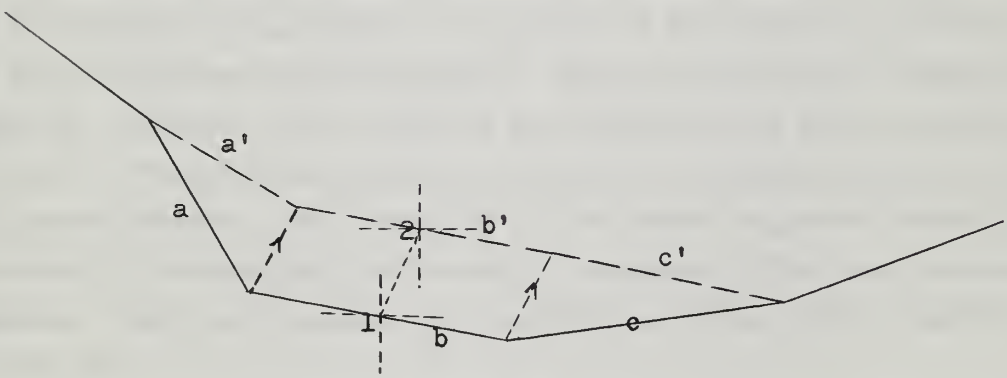


Figure (8)

Sequence a,b,c becomes a',b',c' as tracking cross moves from 1 to 2. Line b' is unchanged in length while a' and c' are altered by the change in the tracking cross position with opposite signs applied.

If the pen knew when it was near the end of a line, it could just move the function point with lines on either side stretching or shrinking. If the line segments were too long, permitting only a gross change, the designer might wish to insert some shorter line segments. Conversely, he might wish to consolidate some line segments. The pointers string organization would be much better than the linear list for this kind of job.

The kind of operation just described is not available on the ESL Console. It could be done either by programming or as a special hardware feature. If it were to be programmed so that the line or point would follow the pen track, the program would have to be a subroutine of DSCOPE, the A Core supervisor for the ESL Display. A cruder version could be written as a B Core program, but the line or point would only be able to relocate on the pen track at unpredictable discrete times according to the processing schedule of the time sharing supervisor. To do this by hardware, a set of active registers, which could be read on command as part of the display list, would have to be provided at the console. These registers would need to be equivalent to two full line generate commands. These registers could then be loaded with the line generate commands on either side of a line or point and incremented in appropriate directions at the same time that the pen track registers are incremented. When the altering of these two lines was complete they could be put back into the normal position in A Core. Figure (9) illustrates the method of incrementing lines as a hardware feature. The transfers to the active registers occupy the locations in the display list normally occupied by the line generate commands. The line commands are temporarily in the active register LR1 and LR2.

It should be pointed out that the present program does not have the facility for changing a complete picture, FP, back into a sequence of temporary pictures, TP's. When there is a facility for manipulating lines in a more sophisticated manner than described in this report, it

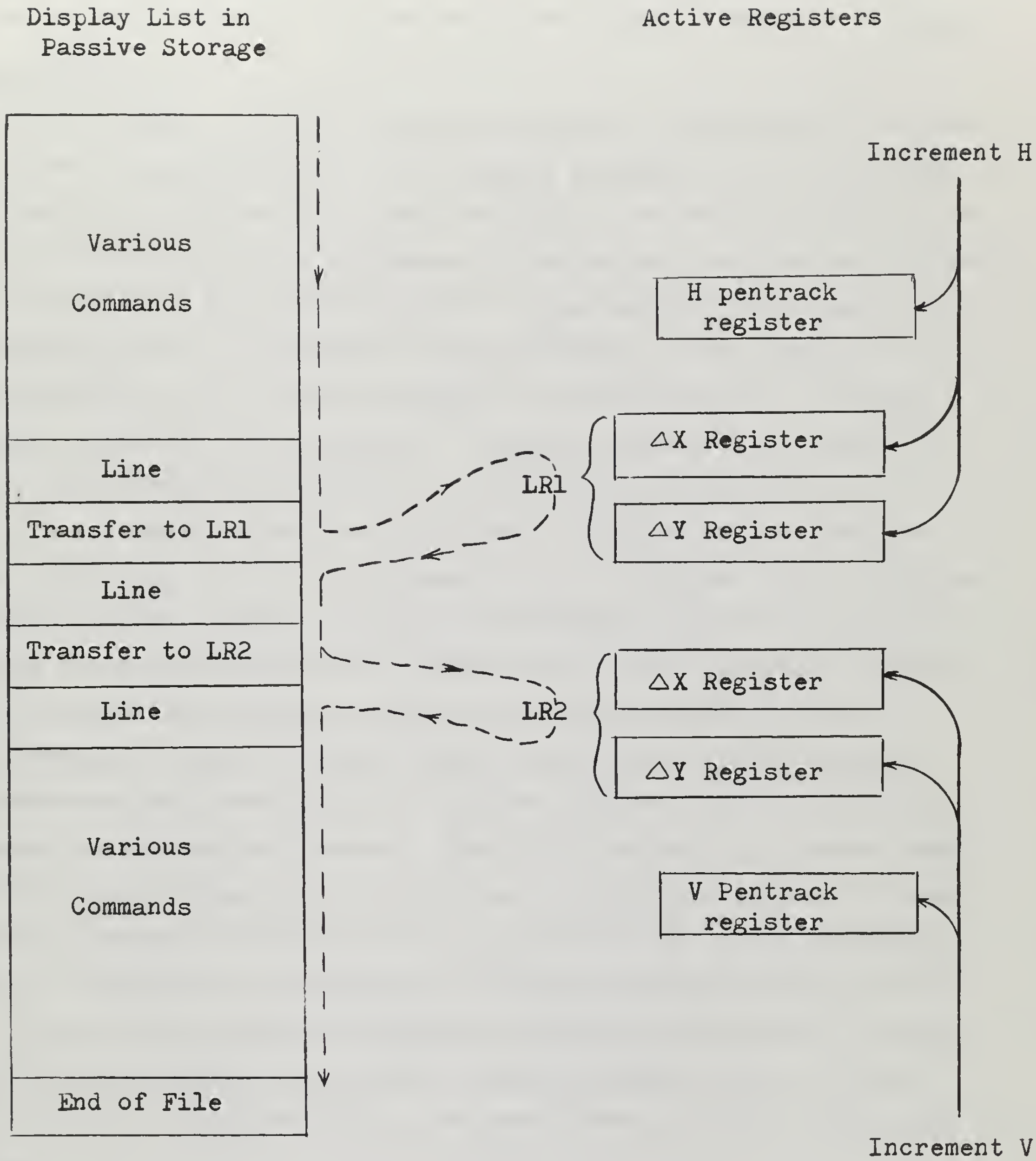


Figure (9)

The use of active Display List registers for moving lines.

will be necessary to write a short routine to replot a complete picture as a sequence of TP's for the purpose of modifying a complete picture.[6]

Working Pictures

A third class of picture, Working Pictures, mentioned earlier, is not presently programmed for. The working picture, WP, is to be made up of complete pictures, FP. The first FP is the reference picture as with the attached pictures displayed on the scope. The working picture is an extension of the attached picture organization but is used only for keeping track of the geometry and properties of the parts of the picture while it is in B core storage or in the disc file. Working pictures as such are never plotted. Working pictures are created by dumping the attached pictures[7] into B Core while the system is in a special mode ready to keep track of them. Attached pictures may be dumped either all at once, in groups, or one at a time. Only attached pictures can form a valid working picture since all of the FP's in an attached chain are oriented to the same axis as the reference picture and have saved the starting positions of the pictures relative to the reference. As the FP's are dumped into B Core, FP(0) becomes the reference and starts a list of pointers which point to the FP's belonging to the working picture. Each FP is stored in a format ready for filing on the disc or for replotting. All the properties of each FP such as weight, scale and name are placed at the top of the saved picture. Any time it is desired to file this working picture on the disc it can be done using the PLEX DUMP routines of Reference 4 or the programmers own system. Similarly, working pictures can be loaded into B Core from the disc file in the same format as above, ready for display.

This kind of organization will be very important in permitting control of picture complexity and display list length by the program and the designer. It will be necessary to control the number of lines

[6] The routines of Robert Polanski will be very helpful for writing picture line manipulation routines.

[7] See Button 12 for description of attached pictures in Section III.

exhibited in order to keep flicker at an acceptable level. It will also be necessary to control detail to prevent the picture from degenerating into a mass of confused lines. In the most obvious way the designer could be permitted to crop out some of the component FP's from the picture by a flick of the pen. The program could be written to exhibit only those FP's which were originally drawn below some selectable scale. Additionally or alternatively the program could examine the FP's in B Core and only plot those which have a certain percentage of the lines over a certain length. There could also be cases where much of the working picture is not actually on the scope but exists somewhere off of it in the "console field". The program could examine the extremes of the component FP's relative to the reference picture and compare them with the relative position of the scope face and plot only those pictures which are presently on the scope face. This would be quite necessary in keeping flicker down. None of this facility for working pictures exists with the present system.

Drawing Within the Arrangement

Once an arrangement plan has been constructed in three dimensions, it forms a framework within which other lines can be drawn to represent such things as piping, ventilation, and wire runs. Several problems may be encountered in attempting such an operation; the main problem would seem to be drawing complexity and clutter. Another difficulty might be requiring small scale in order to see enough of the drawing while at the same time requiring large scale for the drawing operation. Unfortunately, work has not progressed to the point where three-dimensional drawing within a complex picture could be attempted.

However, several kinds of automatic and semi-automatic editing must be available in order to be able to use a cathode ray tube graphical input in the kinds of complex arrangements encountered in ship design. Probably one of the easiest hardware features to implement and use to reduce clutter would be a set of knobs which could control the interval of the field displayed in any of the coordinate directions. If, say, the value of the x coordinate lay outside a certain adjustable range, the scope trace would be turned off. The scope presentation would then appear as a slice of picture in a plane perpendicular to the x axis.

Another difficulty may lie in being able to visually distinguish different types of lines. Colored lines are too expensive. Various patterns of dotted and dashed lines could be realized by turning on and off the beam during the scan. For example, at a scanning rate of 10^4 inches per second, turning the beam on and off at a rate of 100kc/sec would produce a dashed line with 10 dashes/inch.

To be able to draw lines within the three-dimensional space of the arrangement plan requires specifying a point in three dimensions. Two of the dimensions are provided by the light pen position, (h,v); the third dimension may come from a line in the picture, or from a rotation of the picture into a new plane and a repositioning of the tracking cross. The latter method has been used for moving objects up and down the z axis by button 15. The method using depth information obtained from the picture is easier for the draftsman.

To follow a method for specifying a point in three dimensions using picture information, assume the displayed picture has its x, y coordinates coincident with the h, v coordinates of the scope and in the same direction. The procedure then follows:

1) The draftsman brings his tracking cross to a line of the picture causing a "picture part seen" attention to deliver the h and v coordinates of the pentrack and the name of the picture to the B core program.
2) In this example the h, v coordinates are interpreted as x, y coordinates relative to the reference point, of the drawing.
3) Since the starting point of the picture seen is known by the B core program, the program starts adding up the vectors in the picture seen until the relation,

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

is satisfied within some specifiable interval, where (x_1, y_1, z_1) is the beginning of a line segment and (x_2, y_2, z_2) is the end of the line segment relative to the reference point.

4) Then,

$$z = \frac{z_2 - z_1}{x_2 - x_1} (x - x_1) + z_1$$

and the point (x, y, z) is known and can be used as the origin or end of a line.

If the drawing is not in one of the orthogonal picture planes, the same thing can be done but with more algebra. In the general case the operation consists of solving first the equation:

$$(\Delta h, \Delta v, \Delta d) = \underline{M}(\Delta x, \Delta y, \Delta z)$$

where $\underline{M}$ is the rotation matrix, to first get the picture seen in the coordinates of the scope. Then the program searches through the rotated picture to obtain d from h and v. With (h, v, d) known, (x, y, z) is found from the equation:

$$(x, y, z) = M^{-1} (h, v, d).$$

Restricting the plane of the picture to orthogonal planes allows the program to do these two matrix operations by inspection. Provided there is a depth interval selection feature as described previously, it seems that restricting the drawing to the three orthogonal planes would be desirable for the purposes of ship arrangement. First, the draftsman is accustomed to working and thinking in these views and second, an order is imposed on the draftsman which is necessary to keep the layout manageable. This does not in any way confine line segments to the three orthogonal planes. The Working Picture organization would be desirable, but not essential for this type of drawing to be effective.

Computation of Area and Volume

The computation of areas and volumes for ship design is a complex problem toward which much effort is being directed. However, a limited facility for obtaining areas and volumes of specified regions could be made available to the arranger without getting into the more complex general problem. For computing areas the simplest approach would be to provide a framework for computing area based on a quadralateral and let the draftsman specify the extremes of the quadralateral with the light pen. Volume is a little more difficult

but satisfactory information could be obtained using the quadralateral base plus height. If a more refined volume calculation were desired, the space for which volume is required could be quickly filled with regular shapes using a three-dimensional drawing facility. Volumes of prismatoid shapes, having 3 or 4-sided parallel bases, could be correctly interpreted by the computer and drawn on the scope if the 7 or 8 extremes were specified by the light pen. These shapes would remain on the scope until the volume of the filled region is called for. Volumetric centroid could also be computed from these shapes.

Experience With the System

No experience with the complete system as it is described in Section III can be reported since the programs were not completely debugged by the time this thesis was due. Work with pieces of the system give some indications of what kind of difficulties are to be expected. Display list size and flicker will certainly restrict the use of the system until the working picture organization can be implemented to remove from the display list everything that is not on the scope. The program is probably about 20,000 words long including library routines and could grow another 3,000 to 4,000 words by addition of the features mentioned in this section. The entire program has not been completely assembled as yet, but has been used in pieces while sections are being debugged.

There are three categories of computer time used by the system: B Core computation time, swap time, and A Core time for reading the display list, processing attentions and causing real time operations. The B Core computation time per hour of system operation is probably about 1 to 2 minutes per hour. This is a rather poor guess at this time, however. The swap time is a result of the time sharing system swapping programs in and out of B Core as various users are serviced. This time depends on the size of the program, the number of users on the computer and the schedule on which users are serviced. Swap time has been running from 1 to 9 minutes per hour of operation. A Core time used by the display has been estimated fairly accurately to be about 2% of the total computer cycle time.

VI. RECOMMENDATIONS AND CONCLUSIONS

Recommendations

As mentioned, the system is not yet at a state where it can be used to develop an arrangement plan. The features of working pictures, line manipulation and three-dimensional drawing should be added to the system in order for it to handle the complex pictures needed in an arrangement. The ideas for automatic and semi-automatic picture editing should be tried and evaluated. The criteria for picture editing such as scale of original picture and line segment length should be evaluated and new criteria developed and tested. The present library of ship parts is quite small, having only a few simple figures such as a chair, a table and a bunk. A greater variety of standard parts should be added when the three-dimensional drawing feature is added.

Conclusions.

In general, it is felt that the ESL Display is a potentially useful device and that its availability at M.I.T. for use in the computer aided design of ships should be exploited. To do this requires that naval architects at M. I. T. adopt the system through programming to the specific requirements of naval architecture in a manner similar to what has been attempted in this thesis. There will be many problems and setbacks because the ESL Display and supporting software are in the developmental stage, but work toward a specific end, such as ship arrangements, may contribute toward this development in a way that is beneficial to the Navy.

APPENDIX A

SUMMARY OF B CORE SUBROUTINES USED

PLOT (OBJECT, NAME, NAME1*)

This procedure places a block, pointed to by OBJECT, into the display list and gives that object a NAME. If NAME1 is given, the object is inserted into the display list following NAME1, otherwise the object is added at the end of the display list. The asterisk on any argument means that that argument is optional. The format for the object to be plotted is as shown in Figure 10. OBJECT points to the beginning of a block of N + 1 registers. The address of the first register contains the number of registers, N, that will be inserted into the display list. A one in the tag of the top word of the block tells the B Core System to return this block of storage to free storage after the plotting takes place.

CPY (OBJECT, NAME, NAME1*)

The arguments of this procedure have the same meaning as in PLOT except OBJECT is the name of an object already in the display list.

RMV (HERE, THERE*)

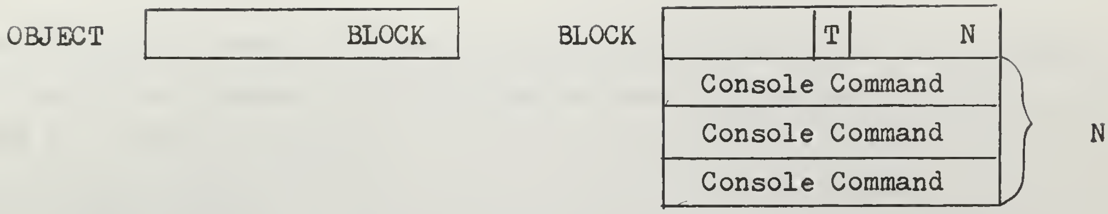
This procedure removes an object or sequence of objects from the display list. HERE and THERE are names of objects in the display list. If two arguments are given, everything from HERE to and including THERE is removed from the display list.

RPL (OBJECT, OLDNAME, NEWNAME*)

This procedure replaces OLDNAME with OBJECT and renames it NEWNAME. If NEWNAME is not given, the name of the new object is OLDNAME. OBJECT may be either of the form of an object as in PLOT or may be already in the display list.

DMP (HERE, THERE*)

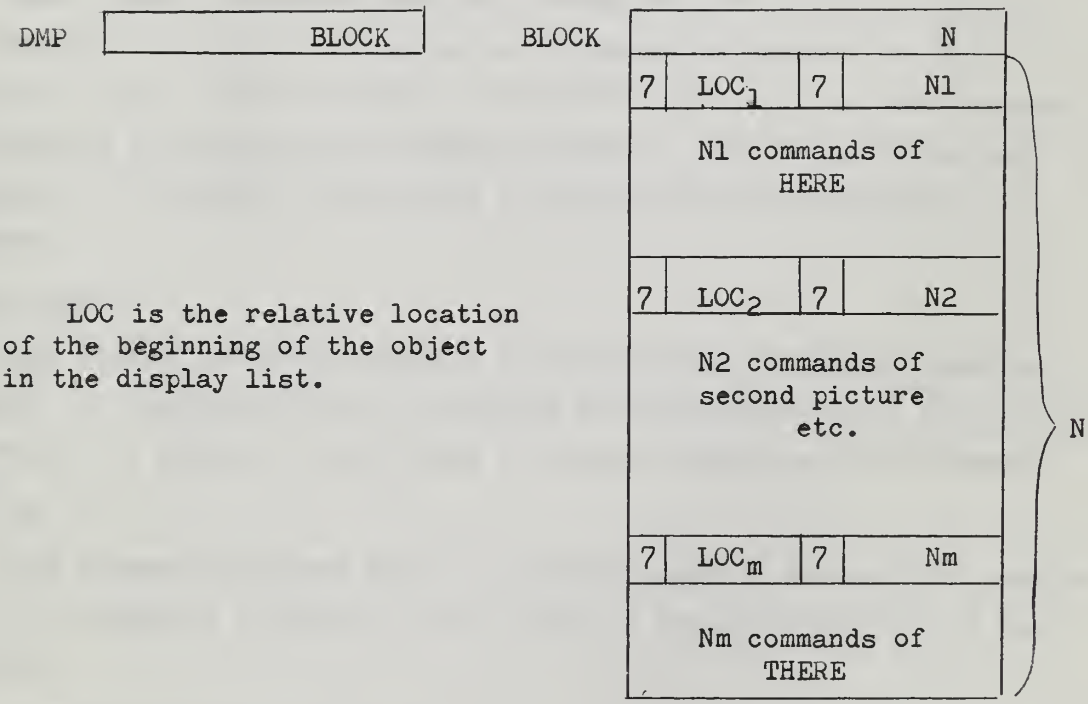
This integer procedure dumps the names from HERE to and including THERE into a block of free storage pointed to by the value of DMP. If only one argument is used, just HERE is dumped. The format of the block is shown in Figure 11. Because of the way the display list is constructed, the last command of the object dumped may be an end of



T is 1 if the B Core system is to return the block of registers to free storage, 0 otherwise.

Figure (10)

Format for plotting objects.



LOC is the relative location of the beginning of the object in the display list.

Figure (11)

Format of DMP(HERE,THERE)

file word or a transfer. These words are indicated by 3's in the prefix and tag of the command. The only way to know whether such a word is there is to check for it.

PLOT (OBJECT, NAME1*)

This is an integer procedure and is an alternate way of plotting objects. The value of PLOT is the location of the display register. NAME1 is optional and means that the object is to be plotted after NAME1. An example of its use is NAME = PLOT (OBJECT), which plots the object at the end of the display list and gives it the name, NAME.

The frequently used objects such as lines, setpoints, rotation matrices and character strings are automatically set up in the proper format for plotting by calling the following integer procedures.

LIN (DELTAX, DELTAY, DELTAZ*)

This integer procedure makes up a standard line generate command. The command will consist of two words if DELTAZ is present and only one if it is not. Even if DELTAX and DELTAY are zero, they must appear as arguments in calling this integer procedure. The value of LIN is a pointer to the block of registers containing the line generate commands.

SINGLE (OCTAL)

This integer procedure makes a standard single character generate command. The argument is the octal code for the letter to be generated. The value is a pointer to the block of storage containing the command.

SETPT (H, V)

This integer procedure makes a standard setpoint command for position H, V. The value is a pointer to the block of registers containing the command.

To build non-standard objects, the following integer procedures were used.

MALIGEC (DELTAX), (DELTAY)

This integer procedure makes a line generate command for the x and y coordinates. The value is the x, y line generate command.

MASEPOI(H,V)

This integer procedure makes a setpoint command. The value is the setpoint command.

NEVIS(COMMAND)

This integer procedure alters the argument, which is LIN or SETPT, so that no trace is generated on the scope. The value is a pointer to the altered command.

Some other procedures and integer procedures are used for special purposes.

SGNON(NUMBER)

This procedure signs the user on to the display console. NUMBER is 1 or 2 to sign on to 1 or 2 consoles.

SGNOFF()

This procedure signs the user off the displays he is using and clears the display list.

SATBUF(NUMBER)

This procedure sets the length of the attention buffer to NUMBER. The attention buffer must be set up if any buttons are to be pushed or lines drawn.

ATTN(ARRAY)

This procedure takes the topmost attention out of the attention buffer and places it in a specified format in ARRAY. Figure (12) shows the format of ARRAY

RLT(FCN,NAME,SPEED,DIR,CONTROL)

This procedure calls for certain real time operations indicated by FCN to be under the control of the knob, globe or pen as indicated by CONTROL. NAME is the name of the object in the display list that will be altered by this real time operation. NAME will always be the name of a setpoint or a matrix. The permissible codes for the various arguments are as follows:

FCN

1 - Rotate about the axis perpendicular to the scope.

2 - Rotate about the h axis.

3 - Rotate about the v axis.

4 - Translate a setpoint up and down, v direction.

No Attentions

1
36

Button Pushed

	N
	0 to 35
Present if pen	H
is tracking.	V

Picture Part Seen by Lightpen

N
38
H
V
Pointer to Display Register

Line Complete

5
41
H
V
ΔX of line
ΔY of line

H,V - position of pentrack on the scope

N - number of words required to specify attention

Figure (12)

Format of array used as argument of ATTN.

5 - Translate a setpoint right and left, h direction.

6 - Magnify by changing rotation matrix uniformly.

7 - Change the Set C control word (See Ref.   for description of Set C word)

0 and 8 - No operation.

9 - Make the setpoint follow the pentrack.

SPEED is any number from 1 to 15. DIR is 1 or 0.

CONTROL

1 - Rightmost knob has changed position.

2 - Center knob has changed position.

3 - Leftmost knob has changed position.

4 - Globe rotated about one of the horizontal axes.

5 - Globe rotated about the vertical axis.

6 - Globe rotated about the other horizontal axis.

7 - Lightpen.

PENH(),PENV()

These integer procedures give the h and v position of the pentrack as their value.

REDIGI(LEFT,RIGHT)

This integer procedure gives as its value the number shown on the decimal switches. The arguments are the switch location numbers that are the beginning and end of the row of consecutive switches that are to be treated as a decimal integer. The switch locations are numbered 1 through 9 from left to right.

DXYALG(ARG)

This integer procedure unpacks a signed component, ARG, of a line generate command into a signed integer. ARG is assumed to be a single register with a 15 bit signed line component in the right 15 bits of the register. The value of DXYALG is a signed integer.

REFERENCES

1. "Operating Manual for the ESL Display Console", by R. H. Stotz and J. E. Ward, Project MAC Technical Memorandum, MAC-M-217, March 9, 1965.

2. "ESL Display Console System Manual", Project MAC Technical Memorandum, MAC-M-201, November 23, 1964.

3. "New B-Core System for Programming the ESL Display Console", by Charles A. Lang, Project MAC Technical Memorandum, MAC-M-216, April 30, 1965.

4. "PLEX Dump and Relocation in the AED-0 System", by Tom Fox, Project MAC Technical Memorandum, MAC-M-198, October 26, 1964.

5. "AED-0 Programmer's Guide" by C. G. Feldman, Project MAC Technical Memorandum, MAC-M-146, April 21, 1964.

6. "A Guide to Algol Programming", by Daniel D. McCracken, John Wiley & Sons, Inc., New York, 1962.

7. "Ship Arrangement System for the ESL Display", by David B. Murton, filed in the Plans Files, Department of Naval Architecture and Marine Engineering, Massachusetts Institute of Technology. 21 May 1965.